ABSTRACT

YOUSEFI, SAEIDEH. MSER-5Y: An Improved Version of MSER-5 with Automatic Confidence Interval Estimation. (Under the direction of James R. Wilson.)

In the steady-state analysis of simulation output, the start-up (or initialization-bias) problem arises when the simulation-generated output process of interest is contaminated by a transient effect that is caused by the analyst’s inability to establish the simulation’s initial condition by randomly sampling from the steady-state probability distribution of system status. In this situation, the simulation passes through a nonstationary phase, ultimately settling down into steady-state behavior. We are typically interested in the parameters of the system in steady-state operation; however the data collected during the nonstationary phase (warm-up period) can be misleading because of the bias in the corresponding simulation responses.

Various methods have been proposed to handle the simulation start-up problem. In this study, we focus on the group of methods that seek to remove the data contaminated with initialization bias by determining an appropriate truncation point, deleting the all the data up to the truncation point, and basing the analysis on the remaining (truncated) data set. The MSER-5 algorithm has received much recent attention because of its simplicity and ease of implementation. MSER-5 uses the truncation point that minimizes the half-length of the classical confidence interval (CI) for the steady-state mean response based on applying the method of nonoverlapping batch means (NBM) to the truncated data set. If the resulting truncation point lies in the latter half of the data set, then the current version of MSER-5 simply fails with an error message that the size of the data set is insufficient. No guidance is provided on how much additional data will be required for MSER-5 to deliver either an acceptable truncation point or acceptable point and CI estimators based on the truncated data set.

The purpose of this research is to develop and evaluate MSER-5Y, an improved version of MSER-5 that is guaranteed to deliver point and CI estimators for the steady-state mean. MSER-5Y also provides a mechanism for sequentially determining the simulation run length required to achieve a prespecified level of precision in estimating the steady-state mean. MSER-5Y seeks the truncation point that is “optimal” in the same sense as for MSER-5; but in the operation of MSER-5Y, the truncation point is constrained to lie in the first half of the data set. Moreover, MSER-5 uses a fixed number of batches (and hence a fixed batch size) in applying the NBM method to the truncated data set so as to deliver the final CI estimator; by contrast, MSER-5Y exploits the von Neumann randomness test to determine the batch size (and hence the batch count) used by the NBM method to deliver the final CI estimator. A carefully selected set of test problems was used to compare the performance of both procedures. MSER-5Y outperformed MSER-5 in almost all the test problems, delivering a better point estimator (i.e., a point estimator with smaller bias and smaller mean squared error) as well as much more reliable CI estimators (i.e., CI estimators with greatly improved coverage probabilities).
MSER-5Y: An Improved Version of MSER-5 with Automatic Confidence Interval Estimation

by
Saeideh Yousefi

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Industrial Engineering

Raleigh, North Carolina

2011

APPROVED BY:

__________________________  ____________________________
Peter Bloomfield           Yahya Fathi

__________________________
James R. Wilson
Chair of Advisory Committee
DEDICATION

To my parents and my sister Simin
BIOGRAPHY

Saeideh Yousefi was born on September 1, 1986, in Tehran, Iran. She was raised by her parents, Mr. Mohammad Sadegh Yousefi and Farkhondeh Jariri, along with her sister Simin.

In the fall of 2004, she entered the Bachelor of Industrial Engineering Program in Tehran University. She graduated with a degree in industrial engineering in 2008. In the spring of 2009, she began her graduate study in industrial engineering at North Carolina State University. She is a member of Alpha Pi Mu, the honor society for Industrial Engineers.
ACKNOWLEDGEMENTS

I would like to thank Dr. James R. Wilson for his encouragement, support, and guidance through the long process of completing this thesis. This research would not have been possible without his kindness, patience, and invaluable assistance. I wish to express my gratitude to Dr. Yahya Fathi for serving on my advisory committee and for his continuous help as the Director of Graduate Programs on various occasions during my graduate study. I would also like to thank Dr. Peter Bloomfield for serving on my advisory committee. Additional thanks are due to Daniel Underwood, who has provided extensive help along the way. Finally, I wish to express my love and gratitude to my parents for their support and endless love through the duration of my studies, and to my sister who has always been there.
# TABLE OF CONTENTS

List of Tables ................................................. vi

List of Figures ............................................... vii

Chapter 1 Introduction ........................................ 1
  1.1 Problem Statement and Research Objectives ............... 2
  1.2 Organization of the Thesis ............................... 4

Chapter 2 Literature Review ................................... 5
  2.1 The Simulation Start-up Problem .......................... 5
  2.2 Review of Previous Literature on the Approaches to Truncation ........................................ 7
  2.3 Observations on the MSER Algorithm from the Literature Review ........................................ 18

Chapter 3 Methodology ........................................ 21
  3.1 The Heuristics under Consideration ....................... 21
    3.1.1 MSER-5 .............................................. 21
    3.1.2 MSER-5Y ............................................. 23
  3.2 Overview of the Experiments ............................... 29
    3.2.1 M/M/1 Queue-Waiting-Time Process with Empty-and-Idle Initial Condition and 90% Server Utilization ........................................ 30
    3.2.2 M/M/1 Queue-Waiting-Time Process with 113 Initial Customers and 90% Server Utilization ........................................ 32
    3.2.3 M/M/1/LIFO Queue-Waiting-Time Process with Empty and Idle Initial Condition and 80% Server Utilization ........................................ 34
    3.2.4 First-Order Autoregressive (AR(1)) Process .......... 36
    3.2.5 AR(1)-to-Pareto (ARTOP) Process .................... 39
  3.3 Performance Measures ..................................... 42

Chapter 4 Results ............................................. 45

Chapter 5 Conclusion .......................................... 61
  5.1 General Conclusions ....................................... 61
  5.2 Contributions ............................................ 62
  5.3 Future Research of the Research ........................... 62

References .................................................. 63

Appendices ................................................ 65
  A1 Matlab Code for MSER-5 algorithm ......................... 66
  A2 Matlab Code for MSER-5Y algorithm ........................ 71
| Table 4.1 | Performance of MSER-5 and MSER-5Y in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition | 46 |
| Table 4.2 | Performance of MSER-5 and MSER-5Y in the $M/M/1$ queue-waiting-time process with 90% server utilization and 113 initial customers | 47 |
| Table 4.3 | Performance of MSER-5 and MSER-5Y in the $M/M/1/LIFO$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition | 48 |
| Table 4.4 | Performance of MSER-5 and MSER-5Y in the AR(1) process with $\rho=0.995$, $X_0=0$ and $\mu=100$ | 49 |
| Table 4.5 | Performance of MSER-5 and MSER-5Y in the ARTOP process | 50 |
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Transient and steady-state density functions for a particular stochastic process (X_1, X_2, \ldots) and initial condition (X_0)</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Flow chart of MSER-5</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Flowchart of MSER-5Y</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>A realization of the M/M/1 queue-waiting-time process with empty-and-idle initial condition and 90% server utilization</td>
<td>31</td>
</tr>
<tr>
<td>3.4</td>
<td>Sample autocorrelation function for an M/M/1 queue-waiting-time process with empty-and-idle initial condition and (\rho = 0.9) based on (N=10,000) observations</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>A realization of the M/M/1 queue-waiting-time process with 113 initial customers and 90% server utilization</td>
<td>32</td>
</tr>
<tr>
<td>3.6</td>
<td>Sample auto-correlation function for an M/M/1 queue-waiting-time process with 113 initial customers and (\rho = 0.9) based on (N=10,000) observations</td>
<td>33</td>
</tr>
<tr>
<td>3.7</td>
<td>Three independent realizations of the transient behavior of the M/M/1 queue-waiting-time process with 113 initial customers and 90% server utilization</td>
<td>33</td>
</tr>
<tr>
<td>3.8</td>
<td>A realization of the M/M/1/LIFO queue-waiting-time process with empty-and-idle initial condition and 80% server utilization</td>
<td>35</td>
</tr>
<tr>
<td>3.9</td>
<td>Sample auto-correlation function for an M/M/1/LIFO queue-waiting-time process with empty-and-idle initial condition and (\rho = 0.8) based on (N=10,000) observations</td>
<td>35</td>
</tr>
<tr>
<td>3.10</td>
<td>A realization of the AR(1) process with (\rho = 0.995), (X_0=0) and (\mu=100)</td>
<td>37</td>
</tr>
<tr>
<td>3.11</td>
<td>Sample auto-correlation function for the AR(1) process with (\rho = 0.995), (X_0=0) and (\mu=100) based on (N=10,000) observations</td>
<td>37</td>
</tr>
<tr>
<td>3.12</td>
<td>Transient behavior of the AR(1) process with (\rho = 0.995), (X_0=0) and (\mu=100)</td>
<td>38</td>
</tr>
<tr>
<td>3.13</td>
<td>A realization of the ARTOP process</td>
<td>40</td>
</tr>
<tr>
<td>3.14</td>
<td>Sample auto-correlation function for the ARTOP process</td>
<td>40</td>
</tr>
<tr>
<td>3.15</td>
<td>Transient behavior of the ARTOP process</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with (X_0 = 0) and (\rho = 0.9) (Sample size 10,000)</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with (X_0 = 0) and (\rho = 0.9) (Sample size 20,000)</td>
<td>52</td>
</tr>
<tr>
<td>4.3</td>
<td>Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with (X_0 = 113) and (\rho = 0.9) (Sample size 10,000)</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with (X_0 = 113) and (\rho = 0.9) (Sample size 20,000)</td>
<td>54</td>
</tr>
</tbody>
</table>
Figure 4.5  Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1/LIFO queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 10,000) ........................................... 56

Figure 4.6  Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1/LIFO queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 20,000) ........................................... 56

Figure 4.7  Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to AR(1) process with $\rho = 0.995$ (Sample size 10,000) .................... 58

Figure 4.8  Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to AR(1) process with $\rho = 0.995$ (Sample size 20,000) .................... 58

Figure 4.9  Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to ARTOP process (Sample size 10,000) ................................. 60

Figure 4.10 Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to ARTOP process (Sample size 20,000) ................................. 60
Chapter 1

Introduction

Computer simulation studies for years have been an important tool in evaluation of various systems. And, they continue to be popular owing to the development of various user-friendly simulation packages with interactive capabilities and also the increased use of object- and logic-oriented programming for traditional simulation modeling.

Simulation by definition is the act of creating a mathematical-logical model of a real system or process and implementing the model on a computer in order to gain insight into the operation of the real system without the limitations imposed by real-world operations. Simulation studies are widely used in physics, astrophysics, chemistry, biology, human systems in economics, psychology, social science, and engineering. In general, a simulation study consists of building a computer model that represents the key characteristics of the system under study, running the model to generate output data, and finally analyzing the output data to understand the behavior of the system. Usually a great deal of effort is needed in building the simulation model and translating it into the relevant software, especially in more complex systems. However, the analysis of the results is just as important; and special attention should be paid in translating the raw simulation observations into useful information about the system.

Traditionally simulation output analysis methods are divided into two different classes, terminating (finite-horizon) simulations and nonterminating (steady-state) simulations. In the former class of output analysis methods, the simulation run has an initial condition and a terminating criterion that are already known to the analyst through previous knowledge or assumptions about the system. In these studies usually multiple runs of the simulation model are made; and since the results of each run are independent from the others, classical statistical methods are used to analyze the resulting independent and identically distributed (i.i.d) random observations accumulated over each run. On the other hand, nonterminating simulations do not have prespecified initial and terminating conditions. Therefore, usually a convenient initial condition is chosen by the analyst; and then the simulation model is run for a long period of simulated time. In some cases a few independent runs of prolonged duration are performed, but the observations within each run will be correlated; and hence the classical statistical methods based
on i.i.d observations are not generally applicable. In steady-state simulations the model is initially in a nonstationary phase; then if the process is stable, it moves toward a steady-state condition in which the underlying probability law “settles down” to its limiting form. We are typically interested in the characteristics of the system in its steady-state condition; however, the observations gathered during the initial period are not typical of steady-state behavior. The challenge here is to minimize the effect of the initial data on the output analysis without losing any useful information. This is called the simulation start-up (or initialization-bias) problem, and it has been a long-standing problem in the field of simulation.

1.1. Problem Statement and Research Objectives

In this study the focus is on the initialization bias or start-up problem in steady state simulations. The different methods present in the literature for dealing with the initialization problem can be categorized into three main groups based on the following approaches:

(a) Determine a run length so large that the initialization bias is “swamped out”;

(b) Set the initial condition close to the “most likely” behavior of the system in steady-state operation to diminish the transient behavior; and

(c) Determine an optimal truncation point (statistics-clearing time); let the model warm up to that point in simulated time; and then delete the warm-up data and base the analysis on the remaining truncated data.

This study is focused on the last method — namely, removal of the data contaminated with initialization bias by specifying an appropriate truncation point. This requires a reliable method for estimation of the effective length of the warm-up period. Ignoring the existence of this period can lead to a significant bias in the final results. On the other hand, the removal of any observations decreases the sample size and thus increases the variance of the relevant sample statistics (for example, the sample mean and variance of the truncated data set). So analysis of the simulation results depends on finding an optimal truncation point up to which we discard the data that is significantly biased by the initial transient.

Various researchers have proposed different methods for finding an optimal warm-up period in a steady-state simulation. Two recent studies by Pasupathy and Schmeiser (2010) and Hoad et al. (2010) provide a reasonably complete bibliography of the different warm-up methods in the literature. The MSER-5 algorithm has received a considerable amount of attention recently because of its simplicity and ease of automation (White 1997; White and Robinson 2010). The basic idea underlying this method is to select a truncation point that minimizes the half-length of the classical confidence interval (CI) for the steady-state mean response that is based on the truncated sample mean and variance as well as the usual critical value of Student’s $t$-distribution. The notion behind this method is easy to understand, the algorithm is quick to automate and run, and it has proved to perform effectively in some limited
simulation experiments with stochastic processes. The latest version of the MSER algorithm, MSER-5, is described in Franklin and White (2008). Despite all the advantages mentioned for this method, some shortcomings of the method have been reported. The main problems with MSER-5 are the following:

(a) In simulation-generated responses with a pronounced initial transient, nonzero skewness or other departures from normality, or significant correlation between successive responses, MSER-5 frequently fails to deliver either a truncation point or a point estimator of the steady-state mean; and

(b) Even when MSER-5 successfully delivers a truncation point together with the truncated sample mean as a point estimator of the steady-state mean, the procedure has no mechanism to deliver a valid CI estimator for the steady-state mean — in particular, the statistical anomalies mentioned in item (a) above generally invalidate the classical CI based on the truncated sample mean and variance as well as the usual critical value of Student’s $t$-distribution.

(c) For the instances where MSER-5 fails or does not deliver a valid CI, there have not been any recommendations on how to continue (restart) the simulation run so as to obtain a data set whose size is sufficient to yield useful point and CI estimators of the steady-state mean.

The objective of this study is to provide an extensive literature review of the current methods for handling the simulation start-up problem, with a special focus on the published work concerning the MSER-5 method. Then, building on the observations made from the literature review, we introduce a modified version of the MSER-5 method, called MSER-5Y, for resolving the start-up problem. The modified method basically uses the same rule for finding the optimal truncation point as MSER-5; however MSER-5Y always delivers a truncation point as well as a (nearly unbiased) point estimator of the steady-state mean and an approximately valid CI estimator of the steady-state mean. Whereas MSER-5 always works with aggregated simulation responses (batch means) having the fixed batch size of 5, MSER-5Y adjusts the final batch size used to construct the CI estimator so that the final batch means pass the von Neumann (1941) test for randomness. Provided that the relative precision of the CI (that is, the ratio of the CI’s half-length to the magnitude of its midpoint) does not exceed 10%, then in our experience the point and CI estimators of the steady-state mean delivered by MSER-5Y are approximately free of initialization bias and are useful in practice. Moreover, in cases where the desired CI is not achieved, MSER-5Y provides a systematic approach for deciding how much additional data is required.

In an extensive experimental performance evaluation, MSER-5 and MSER-5Y are compared in terms of their effectiveness in removing the initial transient for a carefully selected set of test problems with patterns of initialization bias that are typical of many large-scale simulation applications. Four different sample sizes are used to test the effectiveness of the two procedures for large, medium, and relatively small output streams. The performance of the two methods is analyzed in terms of performance measures that are indicative of the accuracy and reliability of the point estimator of the steady-state
mean as well as the CI estimator. A fair comparison of both the methods is achieved by performing
1,000 applications of both MSER-5Y and MSER-5 to the same 1,000 independent realizations of each
test process.

1.2. Organization of the Thesis

The remainder of the thesis is organized as follows: Chapter 2 presents a brief overview of the simulation
start-up problem and the previous literature on this topic, with a special focus on the work published
on the MSER-5 method. At the end of Chapter 2, we present the conclusions made from the literature
review that motivate this study. In Chapter 3 we introduce the methods under consideration—namely
the MSER-5 algorithm as well as the modified algorithm called MSER-5Y; we describe the experiments
used to evaluate the methods; and finally we define the performance measures used in this evaluation.
Chapter 4 presents the results of the experiments. Chapter 5 contains the concluding remarks and
suggestions for future research.
Chapter 2

Literature Review

In this chapter we will give a more detailed description of the start-up problem to provide the reader with background information regarding the start-up problem. Then we continue to summarize the related research efforts with a special focus on the work published on the MSER-5 method with a purpose of introducing the MSER truncation heuristic and the path of its evolution. Then, we present the observations made on the performance of the method based on the literature review upon which we will introduce our modifications to MSER-5 in the next chapter.

2.1. The Simulation Start-up Problem

Simulation studies are primarily conducted in order to obtain reliable information about parameters of interest, which cannot be readily obtained through analytical methods. For the analysis of simulation output, we use several statistical methods. Most of these statistical methods are applied under the assumptions that the simulation output observations are independent and identically distributed. However, many real world processes and simulations are nonstationary and correlated. Consider a stochastic output process from a single simulation run given by $X_1, X_2, X_3, \ldots$. In general, the process can be expressed as \{X_j\} for $j = 1, 2, 3, \ldots$. These successive observations in this process, in general, will neither be independent nor identically distributed; moreover in many simulation-generated responses are not even approximately distributed according to a normal distribution. As a result, conventional statistical methods (such as the usual confidence interval based on Student’s $t$-distribution) cannot be applied directly.

For above defined simulation output process, let the initial condition at the start of the simulation be represented by $X_0$. Let $F_j(x|X_0) = P(X_j \leq x|X_0)$, where $F_j(x|X_0)$ is the transient distribution of the process at time $j$, for initial condition $X_0$. The transient distribution varies for different values of $j$ and $X_0$. If $F_j(x|X_0) \rightarrow F(x)$ as $j \rightarrow \infty$ for all $x$ and for any initial conditions $X_0$, then $F(x)$ is called the steady-state distribution of the output process $X_1, X_2, X_3, \ldots$. Ideally, $F_j(x|X_0) \rightarrow F(x)$ only
as $j \to \infty$; but in practice, there will be a finite time index beyond which these distributions would be approximately the same as each other. Thus, if $d$ is such a point which satisfies this condition, then each observation $X_j$ with index $j > d$ can be said to have been sampled approximately from the steady-state distribution. Also, the individual observations $\{X_j : j = d + 1, d + 2, \ldots\}$ would not be independent, but are assumed to constitute an approximately covariance stationary process. The steady-state distribution $F(x)$ is independent of the initial condition $X_0$, but the rate of convergence of $F_j(x|X_0)$ to $F(x)$ is not. Also, the steady-state distribution is not necessarily a normal distribution. Figure 2.1 illustrates the convergence of the transient probability distributions of the random variable $\{X_j\}$ to the steady-state distribution as $j \to \infty$.

![Figure 2.1: Transient and steady-state density functions for a particular stochastic process $X_1, X_2, \ldots$ and initial condition $X_0$](image)

The simulation for a particular system may be terminating or nonterminating depending on the objectives of the study. A terminating simulation is one for which a specified event $E$ determines the
length of each simulation run. A nonterminating simulation does not have any such event to determine
the end of a replication. The measures of performance of a terminating or nonterminating simulation are
greatly influenced by the state of the system at the beginning of the simulation run, represented by \( X_0 \).
Suppose we are interested in determining a steady-state parameter \( \mu \); i.e. a parameter of the steady-state
distribution \( F(x) \) of a nonterminating simulation. One practical difficulty in estimating \( \mu \) is that we are
trying to estimate the parameters of a distribution \( F(x) \) which would be obtained only as \( j \to \infty \), but in
real-world simulations, we only have a finite set of \( n \) observations from which we try to estimate the
steady-state parameter of interest. Also, it is not always possible to choose the initial condition \( X_0 \) that
is representative of the steady-state behavior of the system. Thus, if we are trying to estimate
\[
\mu = E(X) = \int_{-\infty}^{+\infty} x dF(x),
\]
i.e., the steady state mean of the process, then the sample mean
\[
\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i
\]
will be a biased estimator of \( \mu \) for all values of \( n \). This problem is called as the initialization-bias
problem or the simulation start-up problem.

2.2. Review of Previous Literature on the Approaches to Truncation

Conway (1963)

Conway’s research was one of the first to propose truncation as an efficient method for removing
the effect of the initial transient before reaching statistical equilibrium in simulation. And, he presented
what may be the first formal truncation heuristic. His method suggests selecting the truncation point
as the first point in the sample path that is neither the minimum nor the maximum of the remaining
(truncated) set of data for a single replication of the simulation. Then, Conway suggests repeating this
procedure for several exploratory replications and picking the largest of the resulting points in simulated
time as the final truncation point. His method is referred to as Forward Data Interval Rule in later studies
(White 1995, White 1997). The method has not produced robust and reliable results, usually owing to
the existence of low-amplitude, high-frequency variations in the output sequence that leads to premature
truncation. Two variations to this method are introduced in White (1997) that improve the performance
of Conway’s approach.

Fishman (1973)

Fishman proposes a truncation heuristic referred to as the Crossing-of-the-Mean rule in which the
truncation point is defined as the observation at which the sample path crosses the cumulative mean a
prespecified number of times. White (1997) describes one obvious problem of this method which is determining the appropriate number of crossings. White suggests using the first crossing point as the truncation point, which makes the method specified and easy to apply; however, it is not guaranteed that this is the best choice as the resulting choice for the optimal number of crossings is sensitive to the frequency and magnitude of oscillation in the output sequence.

Fishman (1973) also talks about the effectiveness of truncation methods and suggests that truncating the data results in an inflation of the variance of the truncated sample mean, which is not desired. Fishman concludes there should be balance between mitigating bias through removing initial data against the corresponding increase in the variance. His idea was later used by several researchers to develop heuristics for achieving this balance between bias and variance.

Gafarian, et al. (1978)

Gafarian, et al. present a truncation heuristic very similar to that of Conway (1963). They scan the data using a backward pass beginning with the last observation to find the point that is neither the maximum nor the minimum of all the previously scanned observations. And again similar to Conway they suggest using several exploratory runs and picking the most conservative truncation point as the final truncation point. This method is evaluated in White (1995) and White (1997) and it did not qualify as an effective method for determining the truncation point.

Schruben (1982)

Schruben (1982) present two tests for detecting initialization bias in a sequence of simulation output. In the original test procedure, he uses time series analysis to convert the simulation output to a standardized time series (STS); then he tests the null hypothesis that the converted STS has no bias versus the alternative hypothesis of nonzero initialization bias. Schruben demonstrates the effectiveness of the test by applying it to data from a variety of simulation models. He also presents a modified version of this test, where the output series is separated into two equal parts, and in each part the variance of the sample mean is estimated using Schruben’s STS methodology. The ratio of the two variance estimators is then compared to the appropriate critical value of the \( F \)-distribution with the appropriate degrees of freedom to test the null hypothesis of no initialization bias.

In a later study Schruben et al. (1983) present a family of tests designed to test for specific types of bias— for instance, a quadratic transient mean function. This method, however, requires knowledge of the specific type of bias contained in the output series. Cash et al. (1992), Nelson (1992) and Goldsman et al. (1994) extend these two works and propose variations of the test to generalize Schruben’s procedure. White (2000) presents a comparison of three adaptations of Schruben’s bias detection tests and two variations of the MSER heuristic studied by White (1997).
Welch (1983)

Welch (1983) propose a graphical method that involves multiple replications of the simulation model to estimate the transient mean function with sufficient accuracy so as to estimate the point in simulated time at which the transient mean function “levels off”, finally attaining the value of the steady-state mean. He uses batch-means averaged across the replications as the basis to determine the point at which initialization bias dies out. His method was criticized in White and Minnox (1994) to be ’an elaborate formalization of visualization’ with a number of parameters to be adjusted by the analyst.

Law and Kelton (1983)

Law and Kelton (1983) propose a quantitative approach to truncation based on approximating the transient mean function using piecewise linear regression models. This method requires specification of a large number of parameters and it requires extensive effort from the user with only marginal benefits. Moreover, the procedure is designed for processes with monotonic convergence of the transient mean to the steady-state mean which limits the applicability of the method. In general, White and Minnox 1994 describe this method as “rigorous and precise but only within a prescribed scope of application”.

White and Minnox (1994)

White and Minnox (1994) suggest the first version of heuristic methods that find the optimal truncation point based on minimizing the CI half-length. Their method captures the fundamental logic suggested by Fishman (1972) — balancing reduction in bias against decreased precision (increased variance) owing to decrease in sample size). This balance can be achieved by minimizing the Mean Square Error (MSE) of the truncated sample mean as expressed in the following form. For a simulation-generated output series \( \{X_j: j = 1, 2, ..., N\} \) of length \( N \), the truncated sample mean with truncation point \( d \) (for \( d = 0, 1, \ldots, N - 1 \)) is given by:

\[
X_{N,d} = \frac{1}{N - d} \sum_{j=d+1}^{N} X_j. \tag{2.3}
\]

If \( \mu \) denotes the steady-state mean

\[
\mu = \lim_{j \to \infty} E[X_j] \tag{2.4}
\]

then for a given run length \( N \) and truncation point \( d \), the mean square error of the truncated sample mean \( \overline{X}_{N,d} \) as an estimator of \( \mu \) is given by

\[
\text{MSE}[\overline{X}_{N,d}] = E[(\overline{X}_{n,d} - \mu)^2] = \text{Bias}^2[\overline{X}_{n,d}] + \text{Var}^2[\overline{X}_{n,d}]. \tag{2.5}
\]

However, \( \text{MSE}[\overline{X}_{n,d}] \) is exceedingly difficult to estimate, as it requires estimates of both the bias and variance of \( \overline{X}_{n,d} \). This estimation is in principle made over a theoretically infinite number of output
sequences which is practically impossible. So, instead of selecting a truncation point to minimize the MSE, White and Minnox (1994) propose selecting a truncation point that minimizes the width of the exact 100(1−α) % CI for μ centered on the truncated sample mean, where the confidence level 1−α ∈(0,1).

Stated formally, their optimal truncation point is:

$$d^* = \arg\min_{0 \leq d < N} \frac{z_{\alpha/2}S_{N,d}}{\sqrt{N-d}}$$

(2.6)

where the truncated sample variance corresponding to the truncation point d is given by

$$s_{N,d}^2 = \frac{1}{N-d} \sum_{j=d+1}^{N} (X_j - \bar{X}_{N,d})^2,$$

(2.7)

and $z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2)$ is the $1 - \alpha/2$ quantile of the standard normal distribution. The goal of the approach defined by Equation (2.6) is to mitigate bias by removing initial observations which are far from the truncated sample mean, but only to the extent that this reduction in disparity is sufficient to compensate for the resulting reduction in sample size in the denominator of Equation (2.6).

White and Minnox (1994) then test their method on four queuing simulations. They conclude that the method yields anticipated results on every replication of the experiment based on whether they achieve the anticipated truncation point. They also report an atypical behavior that may occur in an output sequence in which case the truncation is not indicated not by initialization bias per se, but by an unusual initial condition.

**Rossetti and Delaney (1995)**

Rossetti and Delaney (1995) introduced a family of heuristics that utilize queuing approximation to initialize simulation runs stochastically and to establish estimates for the bias and variance of the truncated sample mean within a run. The rule they present is to define the truncation point d as the point in simulated time that minimizes the estimated mean square error of the truncated sample mean. They conclude utilizing queuing approximations wherever possible reduces bias dramatically and produces precise estimates. It is obvious that the use of their method is limited to certain queuing systems, and the assumptions made are not applicable in all processes where such detailed knowledge about the system is not available.

**Rossetti et al. (1995)**

Rossetti et al.(1995) combines the idea of using queuing approximations from Rossetti and Delaney (1995) with White and Minnox’s (1994) heuristic. Rossetti et al. propose incorporating a control reference point computed from an analytical queuing approximation into the heuristic given by White and Minnox (1994). They suggest using the following heuristic for finding the truncation point when trying to estimate $\theta(=\mu)$ from an output sequence $\{X_j : j = 1, 2, ..., N\}$ generated by a steady state simulation:
\[ d^* = \arg \min_{0 \leq d < n-1} \hat{\text{Bias}}^2[\hat{\theta}] + \hat{\text{Var}}[\hat{\theta}] \]  

(2.8)

where \( \hat{\theta} (=X_{n,d}) \) is the estimate of \( \theta \). The estimate of bias is the difference between the truncated sample mean and the approximation for the parameter \( \theta (=\mu) \) that is available from analytical models (\( \theta_a \)). And, the estimator of \( \text{Var}[\hat{\theta}] \) is the usual sample estimator of the variance of the truncated sample mean with the truncation point \( d \); thus we have the estimators

\[ \hat{\text{Bias}}[\hat{\theta}] = X_{n,d} - \theta_a, \]

(2.9)

and

\[ \hat{\text{Var}}[\hat{\theta}] = \frac{\sum_{i=d+1}^{n} (X_i - X_{n,d})^2}{(n-d)(n-d-1)} \]

(2.10)

This suggestion made an improvement in the performance of the heuristic; however, just like Delaney and Rossetti (1995), the application of (2.8) – (2.10) their method is limited to situations in which adequate knowledge about the simulated process is available and an analytical model for \( \theta (=\mu) \) is available.

**White (1997)**

This paper is a revised and enhanced version of White (1995) and it elaborates the idea presented in White and Minnox (1994). The method previously introduced in White and Minnox (1994) is referred to as MCR, the Marginal Confidence Rule, and its performance is examined in the same four queueing systems used in White and Minnox (1994). Additionally, the results of the MCR truncation heuristic is compared to four alternative heuristics: the Null Rule (no truncation), the Forward Data-Interval Rule (Conway 1963), the Backward Data-Interval Rule (Gafarian et al. 1978), and the Crossing-of-the mean Rule (Fishman 1973). Corrections to the latter three methods are also made in the paper to adjust for their inadequacies in certain cases.

The results of White (1997) confirm the significance of the initialization problem in steady-state simulations. Based on his results, the MCR heuristic is found to improve the estimates of \( \mu \) compared with the case where there is no truncation. White (1997) also finds comparable or “modestly superior” performance of MCR compared with each of the heuristics selected from the literature. However, MCR is shown not to be fail-safe. He claims that successful application of the rule requires simulation runs long enough so that they represent steady-state behavior with sufficiently high fidelity. Moreover, inevitably sequences of rare behavior occur and while removing rare but recurring observations may improve point estimates of the mean, truncation of unusual observations may exacerbate underestimation of the steady-state variance of the output, especially if very few runs are made. White (1997) suggest that methods such as batch means continue to be required in developing CI estimates of the mean in
such applications.

Spratt (1998)

Spratt (1998) defines different accepted methods in the literature and also compares their effectiveness in mitigating bias. He investigates three heuristics tests for initialization bias and two variations of the MCR truncation criterion. He uses the term MSER, Marginal Standard Error Rule, instead of MCR. His main contributions are two modifications to the original MCR method defined by White (1997).

Spratt’s first modification is based on the following observation made by Delaney (1995): the marginal standard error criterion can be overly sensitive to observations at the end of the sample path that are close together in value. Spratt’s first modification to the MSER algorithm is designed to do the following:

(a) Avoid the problem of oversensitivity of the algorithm to the observations at the end of the sample path; and

(b) Avoid performing analysis on insufficient data in cases where steady state occurs towards the end of the sample path so that the data set does not contain enough stable data to perform statistical analysis.

The algorithm Spratt proposes is as follows:

Given the simulation output process \( \{X_j : j = 1, 2, \ldots, N\} \),

1. find \( d^* = \arg\min_{0 \leq d < N - 1} \frac{z_{\alpha/2} \text{SN}_d}{\sqrt{N - d}} \);

2. if \( d^* \leq \lfloor N/2 \rfloor \) (where \( \lfloor X \rfloor \) denotes the greatest integer not exceeding \( X \)), then \( d^* \) is the truncation point; and

3. if \( d^* > \lfloor N/2 \rfloor \) then MSER fails because not enough data has been collected.

Spratt’s second modification is based on Schruben’s (1982) suggestion to aggregate the data into nonoverlapping batches of fixed size and then apply a test for initialization bias to the resulting set of batch means. Spratt proposes batching the data into batches of size 5 before applying the MSER rule; and this version of the algorithm is named MSER-5.

Spratt also notes that the comparative studies done on MSER up to 1998 had been based exclusively on data from queuing models. Thus, he defines data sets to represent a wide range of system behaviors in his evaluations.

Based on extensive testing and evaluations, Spratt concludes that MSER-5 is the most attractive general-purpose heuristic for mitigating the initialization bias problem, and the performance of the original MSER is greatly enhanced by batching the observations prior to using the procedure.
White et al. (2000)

The authors of this paper, aim to compare the performance of five well-known truncation heuristics in the literature on the simulation start-up problem. They use MSER and MSER-5 from Spratt (1998) and three variants of bias-detection tests introduced by Schruben (1982). In their work, White, Cobb, and Spratt present the general form of batched MSER as MSER-\(m\) where to truncate an output series \(\{X_j : j = 1, 2, ..., N\}\), instead of applying the MSER rule to the raw output series, MSER-\(m\) uses the series of \(k = \lfloor N/m \rfloor\) batch averages \(\{Z_j\}\):

\[
Z_j = \left(\frac{1}{m}\right) \sum_{i=1}^{m} X_{m(j-1)+i} \quad \text{for} \quad j = 1, \ldots, k. \tag{2.11}
\]

They also suggest that for implementing the MSER and MSER-5 heuristics, we should evaluate the two rules over the initial half of the data set so that the set of candidate truncation points \(d\) is only allowed to include values \(d^* \in \{0, 1, \ldots, \lfloor N/2 \rfloor\}\). Then, in terms of the truncated sample statistics

\[
Z(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} Z_j \tag{2.12}
\]

and

\[
S^2_Z(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} [Z_j - Z(k, d)]^2, \tag{2.13}
\]

if the MSER criterion \(z_{\alpha/2} S_Z(k, d)/\sqrt{k-d}\) is a nonincreasing function of \(d\) at least over the first half \(\{Z_j : j = 1, 2, \ldots, \lfloor N/2 \rfloor\}\) of the data set so that the minimum of the criterion is evaluated exactly at \(d^* = \lfloor n/2 \rfloor\) then they conclude that the sample size \(N\) is not sufficient; and in this situation MSER-5 fails to deliver either a truncation point \(d^*\) or a point estimation of the steady-state mean. This conclusion is different from the recommendation made in Spratt (2000) concerning the criteria for rejecting a truncation point because of excessive truncation (or equivalently, insufficient sample size). More discussion on this topic and how it affects the procedure will be presented later in Section 2.3.

Consistent with Spratt’s results, White, Cobb and Spratt find that MSER-5 improves on the performance of MSER. They report some cases where MSER occasionally shows inconsistency in detecting bias. This behavior is not observed when the data is batched as in MSER-5. Thus, White, Cobb and Spratt suspect that this anomalous behavior is an artifact of the sensitivity of the original MSER procedure to individual observations.

Franklin and White (2008)

Franklin and White (2008) briefly discuss the intuition behind the MSER-5 heuristic. Their discussion is divided in two key notions; one is that MSER-5 optimizes the objective function most desired in simulation studies — namely, a valid confidence interval for the steady-state mean \(\mu\). The other key notion is that MSER-5 provides a reasonable method for determining when a sequence reaches station-
arity. For the first part, they demonstrate how the MSER test statistic (2.4) is related to the classical CI for \( \mu \) and how minimizing the MSER statistic optimizes the estimate of the mean in the sense of minimizing the MSE of the truncated sample mean as an estimator of the steady-state mean. As for the second notion, in order to explore the possibility of MSER-5 being used as a stationarity test, they do not get into theoretical details; instead they look at the similarity in performance between MSER-5 and a well-studied stationarity test, KPSS proposed by Kwiatkowski et al. (1991). Franklin and White assess the similarity of performance between KPSS and MSER-5 by observing the extent to which either the median or mean KPSS truncation point is within 10% of the MSER-5 truncation point.

Their results indicate that MSER-5 and KPSS do not behave in a similar fashion, as the two procedures appear to be “similar” in only roughly half of the cases. Thus, Franklin and White are not able to make any statements about MSER-5’s utility as a stationarity measure.

The other recommendation they make is to use a correlation-adjusted variance estimate, in place of the usual sample variance (2.13). They use the Phillips-Perron correlation adjusted variance estimator. Their results indicate that this adjustment in fact does not yield an improvement in the results.

**Hoad et al. (2010)**

Hoad et al. (2010) carry out an extensive literature review of warm-up methods with the purpose of producing an automated procedure to estimate the warm-up period in simulation output analysis. They evaluate 44 methods presented in the literature. The methods are organized into five categories:

- Graphical methods,
- Heuristic approaches,
- Statistical methods,
- Initialization bias tests, and
- Hybrid methods.

They then define four main criteria according to which they grade all the methods. The criteria defined by Hoad et al. (2010) include the following:

- Accuracy and robustness,
- Ease of automation,
- Generality, and
- Computational speed.
Methods that do not meet the above criteria are rejected, and the remaining methods are short-listed for further testing to find one or more methods that function well. Based on the preliminary testing of the short-listed methods, they report MSER-5 as the most suitable for automation. For further testing of MSER-5 they use artificially created data sets with precisely-defined parameters. These data sets are characterized by a bias function and a steady-state function. The criteria that specify the bias functions are the following: bias length, magnitude (severity), shape and orientation and the three criteria that define the steady-state function are: the variance, error terms and the type of auto-correlation of data. Since there is complete control over the parameters and the true truncation point is precisely known, Hoad et al. go on to study the impact of all these parameters on MSER-5’s behavior. The performance criteria utilized in this study include the following: probability that the CI estimator covers \( \mu \); closeness of the estimated truncation point to the actual point at which the warm-up period “ends” at least for practical purposes; percentage bias removed by truncation; and pattern and frequency of rejections of failures in the method.

They report several interesting observations in the performance of MSER-5 in their results including:

At the end Hoad conclude that MSER-5 is an effective general purpose method for finding a warm-up period in the majority of data sets tested; and finally they recommend creating a framework for MSER-5 to facilitate its incorporation into a software platform for automated output analysis.

White and Robinson (2010)

In this study, White and Robinson once again investigate the reasoning behind why MSER works. This was the topic of an earlier study by Franklin and White (2008). However, in the new study, White and Robinson look at the initialization problem in a different manner. They distinguish between the biasing effect of initialization and autocorrelation.

They use the number in system process \( \{N(t) : t \geq 0\} \) for an M/M/1 queue with server utilization \( \rho \) as an example to give a demonstration of the initialization problem in both the time and frequency domains. They make the following observations.

There are two different transients associated with steady-state simulation output analysis, one associated with the sample mean during which a sufficiently large number of observations must be collected for the sampling distribution of the output process to approximate the true steady-state distribution of the process. The steady-state distribution of the process \( \{N(t)\} \) is:

\[
\lim_{t \to \infty} P_r\{N(t) = n\} = (1 - \rho)^n \rho^n \quad \text{for} \quad n = 0, 1, \ldots, 
\]

(2.14) (which is the geometric distribution with success probability \( 1-\rho \) and mean \( \mu = \rho/(1 - \rho) \) and variance \( \sigma^2 = \rho/(1 - \rho)^2 \)). They observe that regardless of the initial condition \( N(0) = n_0 \) the process \( \{N(t) : t \geq 0\} \) has the same steady-state distribution. And, this distribution is not imitated by the data (even the truncated data that is “free” of initialization bias) until a large enough sample is realized. The other
transient they observed is what they call the state transient during which the state \( N(t) \) varies across an “atypical” range of values, until it reaches the range of values typical of steady-state operation. This is an artifact of initialization and can be avoided with an appropriate choice of initial condition \( N(0) = n_0 \).

Studying the M/M/1 queue with high and low traffic intensities, they look at the effect of autocorrelation on the two transients defined above. They observe that increasing autocorrelation in the values of the states (that is, \( \text{corr} [N(t_1), N(t_2)] \) for \( t_1 \neq t_2 \)) prolongs both transients. They also report that for the same initial condition, the duration of the mean transient is considerably longer than that of the state transient, because of the additional autocorrelation induced by calculating the cumulative mean.

The authors conclude that in resolving the initial transient problem, there is a minimum run length required to reduce the sampling error to an acceptable level. This minimum run length depends on the degree of autocorrelation and the degree of discrepancy between the initial condition and the range of typical state values at steady-state. This minimum run length increases with higher autocorrelation and the rarity of the initial condition.

White and Robinson then apply the MSER-5 algorithm to a data set consisting of time-averaged values of \( N(t) \) computed over time intervals of equal length, comparing the estimate of the steady-state density (2.14) based on MSER-5 with the estimated density based on the raw (untruncated, unbatched) data set. They conclude that the state transient has been removed by MSER-5. However, there still remains a substantial cumulative mean transient that contributes to the bias in the sample mean estimate and that is not solely an artifact of initialization but is also caused by the autocorrelation in the data. The authors conclude that a significantly longer run is required to compensate for the latter effect. They also conclude that the longest unbiased sequence of sample observations is achieved by reference to the minimum bias and MSE rather than by “correct” initialization. By this analysis, the authors reconfirm the idea that minimizing MSE is a reasonable objective for mitigating the problem of initial transient. Later they demonstrate how the MSER-5, \( S_Z(k,d)/\sqrt{k-d} \) statistic is an approximation to the mean square error of the truncated sample mean \( Z(k,d) \) as an estimator of the steady-state mean.

Finally, White and Robinson compare the replication/deletion framework for steady-state simulation output analysis to the single, long run framework. The former approach has the advantage of having uncorrelated sample mean estimates from each replication; however, it leads to a high cost due to deletion of a conservatively estimated warm-up period from each replication. On the other hand, the single, long run identifies and deletes the warm up period only once; however, there is a degree of correlation between the observations within a run. White and Robinson recommend the use of batch means as a common approach to dealing with the autocorrelation problem. They claim that the batch means method can be greatly more efficient than the deletion/replication approach, and they illustrate that by comparing Welch’s (1983) method to MSER-5. Finally, to compute an approximately valid CI for \( \mu \) based on the truncated data set delivered by MSER-5, White and Robinson recommend rebatching the truncated data set (which consists of \( k - d^* = \lfloor N/2 \rfloor - d^* \) batch means with batch size 5) into 20 “new”
batch means with “new” batch size \(\lfloor (k - d^*)/20\rfloor\) and then computing a classical batch-means CI for \(\mu\) from the rebatched, truncated data set. They conclude that using MSER-5 yielded reasonable point and CI estimates of the steady-state mean.

**Pasupathy and Schmeiser (2010)**

Pasupathy and Schmeiser also study the initial transient problem in steady-state studies. They emphasize on the point estimation aspect and argue for using mean square error as the primary statistical criterion for comparing deletion algorithms and show that the MSER statistic is asymptotically proportional to the MSE. Therefore, deleting the data based on the MSER statistic is reasonable, even for autocorrelated processes.

They also suggest two minor modifications to the MSER-5 algorithm. The two new variations are called MSER-LLM and MSER-LLM2. The former picks the optimal truncation point as the left most local minimum of the MSER curve and, the latter picks the left most minimum of the local minima to be the optimal truncation point based on the MSER statistic. They test the two new algorithms and compare them with the original MSER algorithm based on results for M/M/1 and AR(1) processes. Based on the results, they conclude that MSER-LLM and MSER-LLM2 outperform MSER-5 but all three methods perform well when the effective sample size is large. They further conclude that if minimizing MSE is the goal and if the performance measure is the mean, then the MSER statistic is a solid foundation for initial transient algorithms.

**Mokashi, et. al (2010)**

Mokashi et. al conduct a performance comparison of two methods for dealing with the start-up problem, MSER-5 and N-SKART, a procedure developed by Tafazzoli (2009) which is a nonsequential skewness and autoregression adjusted batch-means procedure. Their comparison is based on the M/M/1 queue waiting times process for a queue with 90% server utilization. Their results indicate the outperformance of N-SKART over MSER-5 in this process. For MSER-5 they report that even though the method is appealing intuitively and is much simpler to implement in practice, it suffers from the following disadvantages:

- The point estimates for the steady-state mean exhibit considerable bias, especially for smaller sample sizes;
- The CI coverages delivered by MSER-5 do not conform to the user-specified coverage levels for the small and medium sample sizes;

At the end, to enhance the effectiveness of the MSER-5 algorithm, they recommend combining the heuristic with a procedure that delivers a valid CI estimator in order to achieve a valid CI associated with the point estimator of the steady-state mean.
2.3. Observations on the MSER Algorithm from the Literature Review

The MSER algorithm is indeed an appealing technique in the literature on the simulation start-up problem. The following characteristics can be concluded from the published articles on this method:

- MSER-5 is simple to understand and implement.
- MSER-5 is applicable to a wide range of systems and is a general-purpose method rather than being specific to one particular type of data.
- MSER-5 is easy and inexpensive to automate.
- MSER-5 yields promising results with respect to the performance of both its point and CI estimator of the steady-state mean.
- MSER-5 is applicable to different output sequences, specifying the best truncation point for each such sequence separately.

However, throughout the literature several shortcomings of the MSER method have emerged that require consideration of other, perhaps more complicated, methods or further possible improvements to the MSER-5 method. In this research, the ultimate aim is to propose a modification of MSER-5 with the objective of improving MSER-5’s performance. To do so we have made the following observations that will be used later to develop the enhanced algorithm to address the following issues:

One issue that arises in the application of MSER-5 is the high failure rate of the procedure when applied to output sequences with a pronounced initial transient or responses that exhibit marked departures from normality or a persistent autocorrelation structure. For example, Mokashi et al. (2010) report applying MSER-5 to a sample sequence of 10,000 waiting times from an M/M/1 queue with 90% traffic intensity starting empty and idle. In their analysis, MSER-5 failed in 290 of 1000 replications, approximately 30% of the time. The high failure rate is particularly problematic in the context of steady state simulation analysis, where one typically runs a small number of large-sample-size replications; a failure in a replication can entail high costs and lead to significant loss of information. According to the rule established by Spratt (1998), MSER-5 fails when the truncation point is determined to be in the second half of the data. In such cases, we must conclude that there is insufficient data to support further statistical analysis. Terminating MSER-5 unsuccessfully in this situation is mainly intended to avoid the oversensitivity of the method to final observations as well as cases in which there is really not enough stable data. However, in cases were “final observation” truncation occurs, this results in the discarding of data that is truly representative of the system.

The ideal adaptation of MSER-5 is one that detects cases with insufficient stable data without eliminating valid data representative of the system. To approach this ideal adaptation, we consider the suggestion made by White et al. (2000) to evaluate the MSER statistic only over the first half of the output
sequence. That is, for a batched sample consisting of \( k \) batch means \((k = \lfloor N/5 \rfloor)\), consider truncation point candidates \( d = 1, \ldots, \lfloor k/2 \rfloor \) and reject the sequence if the optimal truncation point is calculated as \( d^* = \lfloor k/2 \rfloor \). Applying the approach of White et al. (2000) to a similar sequence of M/M/1 waiting times considered by Mokashi, et. al (2010), reduced the number of failures from 360 to 6. This high reduction in failures indicates that the vast majority of the failures were actually due to “final observation” truncation rather than insufficient data.

Furthermore, we recommend removing the failure rule of rejecting when the truncation point is evaluated exactly at \( d^* = \lfloor k/2 \rfloor \) assuming that a reasonably long sequence of observations is available that reaches steady-state by the mid point of the data. Here, we are most concerned with the algorithm being able to handle such normally behaving sequences and this failing rule can result in falsely rejecting sequences because of atypical behavior in the sequence that are not an artifact of start-up problem and might lead to the minimum of the MSER-5 criterion being determined at exactly the mid point; and for the same reason mentioned before we are not interested in the discarding of data that are truly representative of the system.

The second issue encountered in applications of MSER-5 is the autocorrelation in the observations from a steady state simulation output sequence. One artifact of this problem is that MSER-5 does not have a mechanism to deliver a valid CI estimator for the steady-state mean in a simulation replication. To account for the autocorrelation in the data, Franklin and White (2008) explore the possibility of using a correlation-adjusted variance estimator in the MSER-5 algorithm instead of the usual naive variance estimator in the numerator of the MSER statistic. However, they conclude that this modification does not yield any improvement in the data. The other research that has addressed the issue of autocorrelation is White and Robinson’s (2010) recommendation to combine the method of batch means with the MSER-5 algorithm to calculate the associated nominal confidence interval based on the rebatched, truncated data based on \( k' = 20 \) batches. White and Robinson report improved results with this extension of MSER-5.

Overall in reviewing the literature, there still seems to be a need for a systematic method to account for autocorrelation in the data in the MSER-5 algorithm. The recommendation we propose here is originated with Fishman (1972) to apply the von Neumann (1941) randomness test to the truncated data to determine an appropriate batch size and then use the batch means that pass the randomness test for building the confidence interval.

Assume that we have observations \( \{X_i : i = 1, 2, \ldots, N\} \). The batch means for \( k = \lfloor N/m \rfloor \) batches of size \( m \) is calculated as:

\[
X_{i,m} = \frac{1}{m} \sum_{j=1}^{m} X_{m(i-1)+j} \quad i = 1, \ldots, k = \lfloor N/m \rfloor.
\]  

(2.15)

The problem here is to determine a final batch size \( m^* \) such that \( k^* = \lfloor N/m^* \rfloor \) final batch means \( X_{1,m^*}, \ldots, X_{k^*,m^*} \), are independent and identically distributed. The von Neumann test uses the null hypothesis \( H_0 : X_{1,m}, \ldots, X_{k,m} \) are i.i.d The test statistic is defined as (see Fishman 1978)
\[ C_k = 1 - \frac{\sum_{i=1}^{k-1} (X_{i,m} - X_{i+1,m})^2}{2 \sum_{i=1}^{k} (X_{i,m} - \bar{X}_k)^2}, \]  
(2.16)

where \( \bar{X}_k = \frac{1}{k} \sum_{i=1}^{m} X_{i,m} \) is the grand average of the batch means.

If \( X_{1,m}, \ldots, X_{k,m} \) are normal, then under \( H_0 \), the test statistic \( C_k / \sqrt{(k-2)/(k^2-1)} \) has a distribution that converges asymptotically \( (k \to \infty) \) to \( N(0,1) \), the standard normal distribution. Therefore in a two-sided test one accepts \( H_0 \) if:

\[ |C_k| \leq z(\alpha/2) \sqrt{(k-2)/(k^2-1)}, \]  
(2.17)

where \( z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2) \) denotes the \( 1-\alpha/2 \) quantile of the standard normal distribution.

Fishman argues that in the context of using \( C_k \) to test for independence among batch means, it is reasonable to treat \( C_k / \sqrt{(k-2)/(k^2-1)} \) as \( N(0,1) \) for \( k \) as small as \( k \geq 8 \). Fishman also describes a procedure to implementing the two-sided test on successively larger batch sizes until \( H_0 \) is accepted. We will use this procedure in Section 2.2 to find the optimal batch size for the truncated data that gives i.i.d. batch means for calculating a valid CI for \( \mu \) based on the truncated data.

Finally, the last observation we have on the MSER-5 method was the need for a systematic method to increase sample size so that MSER-5 can ultimately deliver point and CI estimators with user-specified levels of precision (i.e., CI half-length) and reliability (i.e., CI coverage probability).

In the following chapters, we build on these observations to define a modified version of MSER-5 that considers the above-mentioned issues. The objectives in introducing the modifications are:

1. Defining a systematic approach to deliver CI estimators for the steady-state mean in output series with autocorrelation in the observations.
2. Defining a fail-safe approach that always delivers point and CI estimators of \( \mu \) when needed.
3. Defining a systematic approach to increasing sample size so that MSER-5 can ultimately deliver the desired point and CI estimators.
Chapter 3

Methodology

In this chapter the purpose is to define the following: (a) the heuristics evaluated in this study, namely, MSER-5 and the modified version, MSER-5Y; and (b) the methodology used to assess and compare the performance of the two procedures. Section 3.1 presents an overview of the algorithms for each of the heuristics. Section 3.2 presents a complete description of the experiments used in the evaluation of the methods. Section 3.3 specifies the performance criteria used here to assess the methods.

3.1. The Heuristics under Consideration

3.1.1. MSER-5

MSER-5 as described in Chapter 2, is a simple heuristic for resolving the initialization problem in steady-state simulation output analysis. This is done through selecting the truncation point that minimizes the half-length of the confidence interval about the truncated sample mean. The method was first proposed by White and Minnox (1994) as the MCR (Marginal Confidence Rule). Modifications of this method were presented in White and Robinson (2010), Franklin and White (2008), and Spratt (1998).

The logic in MSER-5 is that in an output sequence \( \{X_1, X_2, ..., X_N\} \), the observations later in the sequence represent the steady-state better than the initial observations gathered during the time the system is warming up; therefore, the initial observations, which are suspected to be further from the steady-state mean, should be truncated. However, removing data from the beginning of the sequence increases the variance of the truncated sample mean computed from the remaining data. The MSER statistic aims at resolving this problem by minimizing the confidence interval half-length about the truncated sample mean. Thus, the initial observations in the output sequence are truncated to the extent that deleting those observations minimizes the half-length of the confidence interval for the steady-state mean based on the remaining output sequence.

The original MSER-5 algorithm suggests that for each candidate data-truncation point in the data set, we should calculate a CI for the steady-state mean \( \mu = \lim_{i \to \infty} E[X_i] \) based on the remaining observations.
beyond the truncation point; and the CI half-length is taken as a measure of the extent to which all the remaining observations are typical of steady-state behavior, where a smaller CI half-length indicates closer conformity to steady-state behavior. So, finally the point that minimizes the CI half-length is chosen as the truncation point.

Spratt (1998) suggests MSER-5, where instead of using the raw data to calculate the MSER statistic, the raw data is batched into nonoverlapping batches of size 5 and the batch means are used to calculate the MSER statistic. So, for the output sequence \( \{X_i : i = 1, ..., N\} \) of size \( N \), the batch means are calculated as

\[
Z_j = \frac{1}{5} \sum_{i=1}^{5} X_{5(j-1)+i} \quad \text{for} \quad j = 1, ..., k = \lfloor N/5 \rfloor,
\]  

(3.1)

where for each real number \( u \), \( \lfloor u \rfloor \) denotes the floor function—i.e., the greatest integer not exceeding \( u \).

Now the basic data items are the batch means \( \{Z_j : j = 1, ..., k\} \). For any truncation point \( d \), the grand average and the sample variance of the data are given by

\[
Z(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} Z_j \quad \text{and} \quad S^2_Z(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} [Z_j - Z(k, d)]^2,
\]  

(3.2)

respectively. Therefore a 100\((1 - \alpha)\)% CI for \( \mu \) has the form

\[
Z(k, d) \pm z_{1-\alpha/2} S_Z(k, d) / \sqrt{k-d},
\]  

(3.3)

where \( z_{1-\alpha/2} \) denotes the \( 1 - \alpha/2 \) quantile of the standard normal distribution. The statistic that MSER-5 tries to minimize is the half-length of the above CI. Since \( z_{1-\alpha/2} \) is a constant, we consider the MSER-5 statistic to be

\[
\text{MSER5}(k, d) = \frac{S_Z(k, d)}{\sqrt{k-d}}.
\]  

(3.4)

This statistic (3.4) is calculated for every candidate truncation point in the range \( 0 \leq d < k - 1 \) to find the argument \( d \) that minimizes Equation (3.4). Spratt (1998) suggests that it is necessary to calculate the statistic over the entire sequence to find the true minimum for the given data set; and if the truncation point is found to lie in the second half of the data, then MSER-5 should report that the sample size is not adequate for statistical analysis and reject that sample path from further analysis. Stated formally, the optimal truncation point is defined as follows:

\[
d^* = \arg\min_{0 \leq d < k-1} \frac{S^2_Z(k, d)}{k-d}; \quad \text{but if} \quad d^* \geq \lfloor k/2 \rfloor, \quad \text{then MSER-5 fails because of insufficient data.}
\]  

(3.5)
If \( d^* < \lfloor k/2 \rfloor \), then MSER-5 delivers the truncated sample mean \( \bar{Z}(k, d^*) \) as the final point estimator of \( \mu \). To compute the associated nominal 100(1 - \( \alpha \))% CI for \( \mu \), White and Robinson (2009) suggest applying the classical method of nonoverlapping batch means (NBM) to the truncated sequence \( \{Z_j : j = d^* + 1, \ldots, k\} \). They suggest using \( k^* = 20 \) “new” batches with the batch size

\[
m^* = \lfloor (k - d^*)/k^* \rfloor. \tag{3.6}
\]

Therefore with the truncation point \( d^* \) and the new batch size \( m^* \), the \( \ell \)th new batch mean is

\[
\bar{Z}_\ell(m^*, d^*) = \frac{1}{m^*} \sum_{j=1}^{m^*} Z_{d^*+(\ell-1)m^*+j} \quad \text{for} \quad \ell = 1, \ldots, k^*; \tag{3.7}
\]

and the corresponding grand average and sample variance of the new batch means are given by

\[
\bar{Z}(k^*, m^*, d^*) = \frac{1}{k^*} \sum_{\ell=1}^{k^*} \bar{Z}_\ell(m^*, d^*) \quad \text{and} \quad S^2_Z(k^*, m^*, d^*) = \frac{1}{k^*-1} \sum_{\ell=1}^{k^*} \left[ \bar{Z}_\ell(m^*, d^*) - \bar{Z}(k^*, m^*, d^*) \right]^2, \tag{3.8}
\]

respectively. Finally an approximate 100(1 - \( \alpha \))% CI for \( \mu \) is

\[
\bar{Z}(k, d^*) \pm t_{1-\alpha/2, k^*-1} \frac{S^2_Z(k^*, m^*, d^*)}{\sqrt{k^*}}, \tag{3.9}
\]

where \( t_{1-\alpha/2, k^*-1} \) denotes the \( 1 - \alpha/2 \) quantile of Student’s \( t \)-distribution with \( k^* - 1 \) degrees of freedom.

It should be noted that White and Robinson (2009) do not clearly state that Equation (3.9) is a general recommendation for calculating the final CI of MSER-5; however, consistent with Mokashi et al. (2010), we use Equation (3.9) for the purpose of defining the most recent version of MSER-5 in the literature.

To summarize, the steps for implementing MSER-5 are depicted in Figure 3.1 for a given simulation output sequence \( \{X_1, X_2, \ldots, X_N\} \) of arbitrary length \( N \).

### 3.1.2. MSER-5Y

Building on the observations made concerning the literature in Section 2.3, we propose an improved version of MSER-5 that is designed to always deliver point and CI estimators of the steady-state mean \( \mu \). Given an output sequence \( \{X_1, X_2, \ldots, X_N\} \) of size \( N \), MSER-5Y computes \( k = \lfloor N/5 \rfloor \) batch means of size 5 according to Equation (3.1) to evaluate the MSER statistic as described in Equation (3.4); however MSER-5Y only evaluates the MSER statistic for candidate truncation points in the range from 0 to \( \lfloor k/2 \rfloor - 1 \). Formally stated, MSER-5Y determines the optimal truncation point \( d^* \) as follows:

\[
d^* = \arg \min_{0 \leq d < \lfloor k/2 \rfloor} \frac{S^2_Z(k, d)}{k-d}. \tag{3.10}
\]
For sample of fixed size $N$, divide the data into $k = \frac{N}{5}$ batches of size 5 and compute the batch means $\{Z_j\}$ from (3.1).

Set truncation point $d \leftarrow 0$

Compute grand average and sample variance of truncated batch means from (3.2).

Compute the test statistic $MSER5(k,d)$ from (3.4).

Increment truncation point, $d \leftarrow d+1$

Is $d < k-1$?

No

Yes

Determine the truncation point $d^*$ that minimizes $MSER5(k,d)$

Is $d^* \geq \frac{k}{2}$?

No

Rebatch data into $k^* = 20$ batches of size $m^*$ and compute the means $\bar{Z}(m^*,k^*)$ from (3.7)

Calculate “new” grand average $\bar{Z}(k^*,m^*,d^*)$ and sample variance $S_z^2(k^*,m^*,d^*)$ from (3.8)

Deliver truncation point $d^*$, mean estimator $\bar{Z}(k^*,m^*,d^*)$ and CI estimator (3.9)

End

Figure 3.1: Flow chart of MSER-5

The point estimator of $\mu$ is

$$Z(k,d^*) = \frac{1}{k-d^*} \sum_{j=d^*+1}^{k} Z_j. \quad (3.11)$$

Then to compute the $100(1 - \alpha)$% CI for $\mu$, MSER-5Y applies the von Neumann randomness test (see Section 2.3) to the truncated data $\{Z_j : j = d^* + 1, d^* + 2, \ldots, k\}$ to find a “new” batch size $m^*$ for which the “new” batch means are (approximately) independent. We use the procedure introduced in Fishman (1978) to implement the von Neumann test on successively larger batch sizes until the test is
finally passed. To do that, we set the initial batch size $m$ as 1, and calculate the number of batches $k'$ accordingly:

$$m \leftarrow 1 \quad \text{and} \quad k' \leftarrow \lfloor (k - d^*) / m \rfloor.$$  

(3.12)

Then we calculate the corresponding batch means

$$W_j(m) = \frac{1}{m} \sum_{l=1}^{m} Z_{d^* + (j-1)m + l} \quad \text{for} \quad j = 1, 2, ..., k'.$$

(3.13)

To compute the test statistic for the von Neumann randomness test, we compute the following sample statistics from the current set of batch means (3.13): (a) the grand average,

$$\bar{W}(k', m) = \frac{1}{k'} \sum_{j=1}^{k'} W_j(m);$$

(3.14)

(b) the variance

$$S^2_{W}(k', m) = \frac{1}{k' - 1} \sum_{j=1}^{k'} \left[ W_j(m) - \bar{W}(k', m) \right]^2;$$

(3.15)

and (c) the mean square successive difference,

$$\text{MSSD}_W(k', m) = \frac{1}{k' - 1} \sum_{j=1}^{k'-1} \left[ W_{j+1}(m) - W_j(m) \right]^2.$$

(3.16)

Then the von Neumann test statistic is

$$C_{k'} = 1 - \frac{\text{MSSD}_W(k', m)}{2S^2_{W}(k', m)}.$$

(3.17)

The batch means with the current batch size $m$ fail the randomness test at the level of significance $1 - \beta \in (0, 1)$, if

$$\left| C_{k'} / \sqrt{\frac{k' - 2}{(k')^2 - 1}} \right| > z_\beta;$$

(3.18)

otherwise, the current batch means pass the randomness test, and MSER-5Y goes to the final CI calculation with final batch size $m^* \leftarrow m$ and final batch count $k^* \leftarrow k'$. We choose a significance level of $1 - \beta = 0.20$ for the von Neumann test.

If the test with the current batch means fail the randomness test, then the randomness test is reapplied with progressively increasing batch sizes until the batch means are finally determined to be approximately independent of each other. We increase the batch size by a factor of 1.2 at every iteration,
\[ m \leftarrow \lceil 1.2m \rceil, \quad k' \leftarrow \lfloor (k - d^*)/m \rfloor; \quad (3.19) \]

and provided \( k' \geq 10 \), MSER-5Y reperforms equations (3.14)–(3.18) with the updated batch size and batch count. The minimum batch size is considered as 10. Therefore if we do not find at least 10 independent batch means, then MSER-5Y just sets the final batch count as 10 and computes the CI based on the minimum of 10 batches,

\[ k^* \leftarrow 10, \quad m^* \leftarrow \lfloor (k - d^*)/k^* \rfloor. \quad (3.20) \]

For the final set of \( k^* \) batches each of size \( m^* \), the final batch means are calculated from Equation (3.13), and sample statistics for the final batch means are calculated according to (3.14) and (3.15) for the grand average and the sample variance of the batch means, respectively. The final 100\((1 - \alpha)\)% CI estimator for \( \mu \) is then calculated as follows:

\[ Z(k, d^*) \pm \frac{t_{\alpha/2, k^* - 1} \cdot S_W(k^*, m^*)}{\sqrt{k^*}}. \quad (3.21) \]

Note that for the point estimator of \( \mu \) in (3.21), we use \( Z(k, d^*) \) rather than \( W(k^*, m^*) \) because \( Z(k, d^*) \) is the sample mean of all observations in the truncated data set \( \{ Z_j : j = d^* + 1, \ldots, k \} \), whereas \( W(k^*, m^*) \) does not include any observations in the partial batch \( \{ Z_j : j = k^* m^* + 1, \ldots, k \} \) at the end of the truncated data set when \( k^* m^* < k \).

The final step in MSER-5Y is to determine whether the CI estimator (3.21) has acceptable precision; and if the precision is unacceptable, then we must determine how much additional data is required to ensure acceptable precision. We use a general rule to evaluate the delivered CI based on the recommendation made by Heidelberger and Lewis (1983). They suggest that for a CI to be approximately valid, a relative precision of at most 10% should be obtained, even though that does not guarantee adequate CI coverage. We use the same recommendation here; however this can be replaced by any user-specified requirement based on the level of precision desired. If the precision requirement is not satisfied, then MSER-5Y estimates the total number of batches of the current batch size that are needed to satisfy the precision requirement. The relative precision of the CI (3.21) is

\[ R = \frac{t_{\alpha/2, k^* - 1} \cdot S_W(k^*, m^*)}{\sqrt{k^*}} / |Z(k, d^*)|; \quad (3.22) \]

and \( R^* = 0.10 \) is the target relative precision based on Heidelberger and Lewis’s (1983) recommendation. To achieve the precision requirement

\[ R \leq R^*, \quad (3.23) \]

\[ 26 \]
we estimate that the number of required batches with batch size $m^*$ is given by

$$k^{**} \leftarrow \left\lceil \left( \frac{R}{R^*} \right)^2 k^* \right\rceil,$$

(3.24)

where for each real $u$, we let $\lceil u \rceil$ denote the ceiling function—i.e., $\lceil u \rceil$ is the smallest integer not less than $u$.

Figure 3.2 depicts the flowchart of the MSER-5Y algorithm. We will evaluate the point and CI estimators delivered by MSER-5 and MSER-5Y for a wide range of processes in the next chapter. The next section is dedicated to describing the experiments used in the evaluation of the two truncation methods.
For sample of fixed size $N$,
Divide the data into $k = \left\lfloor \frac{N}{5} \right\rfloor$ batches of size 5 and compute the batch means $\{Z_j\}$ from (3.1).

Set truncation point $d \leftarrow 0$

Compute grand average and sample variance of truncated batch means from (3.2)

Compute the test statistic $MSER-5Y(k,d)$ from (3.4)

Increment truncation point $d \leftarrow d+1$

Yes

Is $d \leq \frac{L}{2}$ ?

No

Determine the truncation point $d^*$ that minimizes $MSER-5Y(k,d)$

Deliver truncation point $d^*$, point estimator $Z(k,d^*)$

In the truncated series of size $(k - d^*)$, set batch size $m$ and batch count $k'$ from (3.12)

Reached min batch count (10)?

Yes

Set batch count as minimum batch count (10)

No

Compute "new" batch means $W_{(m)}$ from (3.13)

Compute von Neumann test statistic from (3.14)-(3.17)

Yes

Randomness test passed?

No

Increase batch size; deflate batch count using (3.19)

Deliver CI estimator (3.21)

Compute Relative precision from (3.23)

Determine how many more batches needed

Yes

Is $r$ acceptable?

No

End

Figure 3.2: Flowchart of MSER-5Y
3.2. Overview of the Experiments

The test problems used in this research are carefully selected to provide the following: (a) relatively extreme examples of nonnormal, correlated simulation output processes that in most cases are contaminated by initialization bias; and (b) test problems that are more nearly typical of practical applications. The five processes used here as test problems include:

(a) Queue waiting times for the M/M/1 queue with a server utilization of 90% and an empty-and-idle initial condition;

(b) Queue waiting times for the M/M/1 queue with a server utilization of 90% and 113 initial customers;

(c) Queue waiting times for the M/M/1/LIFO queue with a server utilization of 80% and an empty-and-idle initial condition;

(d) The AR(1) process with autoregressive parameter value of 0.995, steady-state mean of 100, steady-state variance of 10.01, and an initial condition of zero;

(e) The Autoregressive-to-Pareto process obtained by transformation of a stationary version of the AR(1) process.

All these test cases have steady-state parameters (such as the steady-state mean) that can be obtained through analytical procedures; therefore, each test process can be used to compare the performance of MSER-5 and MSER-5Y with respect to their accuracy and precision in estimating the steady-state mean. Also, the stochastic behavior exhibited by the above test problems are typical of many steady-state simulations and will enable a performance comparison of the two methods. In this research, 1000 independent replications of each test problem were generated. We included sample sizes of $N=10,000$, 20,000, 50,000 and 200,000 in an attempt to characterize the performance of the methods in small, medium and large sample sizes. A description of the test problems is provided in the following sections.
3.2.1. M/M/1 Queue-Waiting-Time Process with Empty-and-Idle Initial Condition and 90% Server Utilization

For the first test problem we consider the waiting times from an M/M/1 queue. The M/M/1 queue is one of the simplest queuing systems that exhibit interesting transient behavior. It consists of a one server with exponentially distributed service times \( \{B_j\} \) having mean \( \mu_B = 1.0 \) time units. The customers arrive in the system according to a Poisson process with rate \( \lambda = 0.9 \) customer per unit time. This means that customer interarrival times \( \{A_j\} \) are randomly sampled from an exponential distribution with mean \( \mu_A = 1/\lambda = 1.111 \) time units. Therefore the steady-state server utilization is

\[
\rho = \frac{\lambda}{\mu} = \frac{\mu_B}{\mu_A} = 0.9.
\]  

A queue forms when customers arrive in the system and do not find the server idle. The response variable we are interested in the system is the time each customer spends in the queue. Let \( X_j \) denote the waiting time in the queue for the \( j \)th customer for \( j = 1, \ldots, N \), where we let \( N \) take the values 10,000, 20,000, 50,000 and 200,000. The system starts in empty-and-idle state, meaning the queue is empty and the server is idle. As a result \( X_1 = 0 \) on every replication of the process \( \{X_j\} \). Then the process transitions through a warm-up period and reaches steady-state asymptotically as the run length \( N \to \infty \). The expected steady-state waiting time in queue is given by

\[
\mu = \lim_{j \to \infty} E(X_j) = \frac{\rho}{1 - \rho} = 9.0 \text{ time units}
\]  

The purpose of using this experiment is that even though the warm-up period in this process is considered to be relatively short, the process exhibits a strong autocorrelation structure, with the autocorrelation function for the waiting time decaying slowly with increasing lags. The M/M/1 queue waiting times at steady state have a probability distribution that has a nonzero probability at zero and an exponential tail. This results in a slow convergence of the batch means to the normal distribution with increasing batch size.

A single realization of the waiting times in an M/M/1 queue with traffic intensity 90% starting empty-and-idle is depicted in Figure 3.3 for sample size of 10,000. The corresponding sample autocorrelation function is depicted in Figure 3.4.
Figure 3.3: A realization of the M/M/1 queue-waiting-time process with empty-and-idle initial condition and 90% server utilization

Figure 3.4: Sample autocorrelation function for an M/M/1 queue-waiting-time process with empty-and-idle initial condition and $\rho=0.9$ based on N10,000 observations
3.2.2. M/M/1 Queue-Waiting-Time Process with 113 Initial Customers and 90% Server Utilization

The second test process we have chosen for evaluating our truncation methods, similar to the first test process, is an M/M/1 queue with the same interarrival time and service time processes, therefore we have the same server utilization of 90% for this test process. However, this time we let the initial condition be a rare value, in this case 113 customers, a value that insures that the expected queue waiting time for the first regular customer to arrive after time 0 is 10 standard deviations above the steady-state mean (Mokashi 2010). As the definition of steady state suggests, both this system and the one in the first test converge to the same steady-state parameters regardless of the initial condition. However, the M/M/1 queue waiting time process with 113 initial customer is expected to have a more pronounced initial transient with a much longer duration than that of the first test process.

Figure 3.5 depicts a single realization of the process, and Figure 3.6 shows the corresponding sample autocorrelation function. Figure 3.7 depicts the transient behavior of the process for three independent replications.

Figure 3.5: A realization of the M/M/1 queue-waiting-time process with 113 initial customers and 90% server utilization
Figure 3.6: Sample auto-correlation function for an M/M/1 queue-waiting-time process with 113 initial customers and $\rho=0.9$ based on $N=10,000$ observations.

Figure 3.7: Three independent realizations of the transient behavior of the M/M/1 queue-waiting-time process with 113 initial customers and 90% server utilization.
3.2.3. M/M/1/LIFO Queue-Waiting-Time Process with Empty and Idle Initial Condition and 80% Server Utilization

The next test process under study is the sequence of waiting times for the M/M/1/LIFO process. Arrivals follow a Poisson process with arrival rate of $\lambda=0.8$ so that the corresponding interarrival times $\{A_j\}$ are randomly sampled from an exponential distribution with a mean of $\mu_A = 1/\lambda = 1.25$ time units. The service times $\{B_j\}$ are randomly sampled from an exponential distribution with a mean of $\mu_B = 1.0$ time units. Customers in the queue are served in last-in-first-out (LIFO) order, and the system starts in an empty-and-idle initial condition. Thus, the steady-state server utilization, $\rho$, and steady-state mean waiting time, $\mu$, of the system are calculated as

$$\rho = \lambda \mu_B = \mu_B / \mu_A = 0.8$$

and

$$\mu = \left( \frac{\rho}{1 - \rho} \right) \mu_B = 4$$

respectively.

This process is particularly interesting because of its special autocorrelation function structure. The M/M/1/LIFO queue-waiting-time process, unlike most test processes used in the literature, does not demonstrate a geometrically decaying behavior in its autocorrelation function. Also in steady-state operation, batch means computed from the waiting times are highly skewed—even for batch sizes that are sufficiently large to ensure the batch means are nearly uncorrelated. A realization of the process is depicted in Figure 3.8; and Figure 3.9 depicts the sample autocorrelation function for this realization.
Figure 3.8: A realization of the M/M/1/LIFO queue-waiting-time process with empty-and-idle initial condition and 80% server utilization.

Figure 3.9: Sample auto-correlation function for an M/M/1/LIFO queue-waiting-time process with empty-and-idle initial condition and $\rho=0.8$ based on $N=10,000$ observations.
3.2.4. First-Order Autoregressive (AR(1)) Process

For this test process a sequence of \( \{X_j\} \) is generated according to the following relation:

\[
X_j = \mu_X + \rho(X_{j-1} - \mu_X) + \varepsilon_j \quad \text{for} \quad j = 1, \ldots, N. \tag{3.29}
\]

where the autoregressive parameter \( \rho = 0.995 \), the steady-state mean \( \mu_X = 100 \) and the error terms \( \{\varepsilon_j : j = 1, 2, 3, \ldots\} \) are randomly sampled from a \( N(0, 1) \) distribution. The initial condition for this process is set as \( X_0 = 0 \). The steady-state distribution of the process \( \{X_j\} \) has a mean of \( \mu_X = 100 \) and standard deviation of \( \sigma_X = 10.0125 \).

This test process was selected for several reasons. First of all, with the initial condition of \( X_0 = 0 \) ensures that the expected value of the first observation after time 0 is approximately 10 steady-state standard deviations below the steady-state mean. Therefore, the process is ensured to have a pronounced negative initialization bias. Secondly, although this process is normal, it has a pronounced autocorrelation structure that severely distorts the behavior of the classical method of batch means (Tafazzoli et al. 2011).

A realization of this process is depicted in Figure 3.10, and Figure 3.12 depicts the transient behavior of this process for three independent realizations each of length \( N = 10,000 \). Figure 3.11 shows the sample autocorrelation function corresponding to the realization of Figure 3.10.
Figure 3.10: A realization of the AR(1) process with $\rho=0.995$, $X_0=0$ and $\mu=100$

Figure 3.11: Sample auto-correlation function for the AR(1) process with $\rho=0.995$, $X_0=0$ and $\mu=100$ based on $N=10,000$ observations
Figure 3.12: Transient behavior of the AR(1) process with $\rho=0.995$, $X_0=0$ and $\mu=100$
3.2.5. AR(1)-to-Pareto (ARTOP) Process

The AR(1)-to-Pareto (ARTOP) process is generated from an underlying AR(1) process \( \{Z_j : j = 1, 2, \ldots\} \) with autoregressive parameter \( \rho = 0.995 \) and white noise variance of \( \text{Var} (\epsilon_j) = 1 - \rho^2 = 0.9975 \times 10^{-2} \) so that in steady-state the \( \{Z_j\} \) are standard normal random variables with lag-one correlation 0.995.

The corresponding observations \( \{X_j : j = 1, 2, \ldots\} \) of the target ARTOP process are generated from the Pareto c.d.f.,

\[
F_X(x) \equiv \Pr\{X \leq x\} = \begin{cases} 1 - (\xi / x)^\psi, & x \geq \xi, \\ 0, & x < \xi, \end{cases} \quad (3.30)
\]

where \( \xi > 0 \) is a location parameter and \( \psi > 0 \) is a shape parameter.

To generate the ARTOP process \( \{X_j : j = 1, \ldots, N\} \) we use the following steps:

First generate the \( \{Z_j\} \) according to the relation

\[
Z_j = \rho Z_{j-1} + \epsilon_j \quad \text{for} \quad j = 1, \ldots, N, \quad (3.31)
\]

where \( \{\epsilon_j : j = 1, \ldots, N\} \) \( \text{i.i.d.} \sim N(0, \sigma^2) \) is a white-noise process with variance

\[
\sigma^2 = \sigma_Z^2 (1 - \rho^2) = 1 - \rho^2. \quad (3.32)
\]

This process is used to generate a set of correlated random variables \( \{U_j = \Phi(Z_j) : j = 1, \ldots, N\} \) whose marginal distribution is Uniform(0, 1), where

\[
\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2} \, d\zeta \quad \text{for all real } z \quad (3.33)
\]

is the \( N(0, 1) \) c.d.f. The correlated sequence of random numbers \( \{U_j : j = 1, \ldots, N\} \) is supplied as input to the inverse of the Pareto c.d.f. as described in Equation (3.30). Finally, the ARTOP process \( \{X_j : j = 1, \ldots, N\} \) is generated as follows:

\[
X_j = F_X^{-1}(U_j) = F_X^{-1}[\Phi(Z_j)] = \frac{\xi}{[1 - \Phi(Z_j)]^\psi} \quad \text{for} \quad j = 1, \ldots, N. \quad (3.34)
\]

We chose \( \psi = 2.1 \) and \( \xi = 1.0 \) as the parameters of the Pareto distribution. Furthermore, we chose the initial condition \( Z_0 = 3.4 \); therefore, we have \( X_0 = F_X^{-1}[\Phi(Z_0)] = 43.569 \), which induces a pronounced positive bias in the test process because the steady-state mean of the ARTOP process is given by

\[
\mu = \lim_{j \to \infty} E[X_j] = \psi \xi (\psi - 1)^{-1} = 1.9091. \quad (3.35)
\]

Clearly, this process exhibits pronounced correlation among successive observations as well as severe...
nonnormality. Figure 3.13 shows a single realization of the ARTOP process; and Figure 3.14 shows the sample autocorrelation function associated with this realization. Figure 3.15 depicts three independent realization of the ARTOP process.

Figure 3.13: A realization of the ARTOP process

Figure 3.14: Sample auto-correlation function for the ARTOP process
Figure 3.15: Transient behavior of the ARTOP process
3.3. Performance Measures

For the purposes of comparison, the test processes described in the previous sections are each generated for 1000 independent replications. We are interested in the average performance of MSER-5 and MSER-5Y over these independent replications, where we have used the method of common random numbers (Law 2007) to ensure that for each realization of each test process, both MSER-5 and MSER-5Y are applied to the same simulation-generated time series. This approach ensures the sharpest possible comparison of MSER-5 and MSER-5Y in terms of each procedure’s ability to deliver point and CI estimates of the steady-state mean that are approximately free of initialization effects. The effectiveness of the truncation methods in estimating the steady-state characteristics are measured in terms of two types of performance measures:

(a) Measures that characterize the properties of the confidence interval estimator delivered by each method; and

(b) Measures that characterize the properties of the point estimator delivered by each method.

Using nominal coverage probabilities of 90% and 95%, we computed the following statistics from group (a) above to evaluate the effectiveness of MSER-5 and MSER-5Y:

- CI coverage measures the probability that the true steady-state mean $\mu$ falls within the calculated confidence limits;
- CI relative precision is the ratio of the CI half-length to the magnitude of the corresponding point estimator as in Equation (3.22);
- CI half-length measures the precision of the CI estimator; and
- Variance of the CI half-length measures the variability of the CI estimator.

From group (b) above, we computed the following performance measures to evaluate the point estimators delivered by MSER-5 and MSER-5Y:

- The overall average truncated sample mean is the grand average of the truncated sample means computed over all 1,000 replications of each test process;
- The bias measures the systematic deviation of the point estimator away from $\mu$ and is estimated by the difference between the overall truncated sample mean above and the steady-state mean $\mu$;
- The variance of the truncated sample mean measures the random variation around the point estimator’s expected value and is estimated by the sample variance of the 1,000 truncated sample means for each test process; and
• The mean squared error is used as a measure of the overall accuracy of the truncated sample mean as an estimator of $\mu$, as it combines both the squared bias and the variance of this estimator.

To summarize, for both MSER and MSER-5 certain key statistics are calculated over the $Q$ replications of each test process, where on replication $q$ (for $q = 1, \ldots, Q$), we let $\bar{Z}_q(k, d^*_q)$ denote the final truncated sample mean based on the truncation point $d^*_q$ so that we have the following performance measures from group (b) above:

$$\hat{\mu}_Q = \frac{1}{Q} \sum_{q=1}^{Q} Z_q(k, d^*_q)$$  \hspace{1cm} (3.36)

is the overall average of the truncated sample means, which estimates the expected value $E[\bar{Z}_q(k, d^*_q)]$;  

$$\hat{\text{Var}}_Q = \frac{1}{Q-1} \sum_{q=1}^{Q} [Z_q(k, d^*_q) - \hat{\mu}_Q]^2$$  \hspace{1cm} (3.37)

is the sample variance of the truncated sample means, which estimates the variance $\text{Var}[\bar{Z}_q(k, d^*_q)]$ of the truncated sample mean; and  

$$\hat{\text{MSE}}_Q = \frac{1}{Q} \sum_{q=1}^{Q} [Z_q(k, d^*_q) - \mu]^2$$  \hspace{1cm} (3.38)

estimates the mean squared error of the truncated sample mean,  

$$\text{MSE} \equiv E\{[\bar{Z}_q(k, d^*_q) - \mu]^2\}. \hspace{1cm} (3.39)$$

We are also interested in the bias of the truncated sample mean,  

$$\text{Bias} \equiv E[\bar{Z}_q(k, d^*_q)] - \mu; \hspace{1cm} (3.40)$$

and we estimate the magnitude (absolute value) of this quantity indirectly through the fundamental relation  

$$\text{MSE} = \text{Bias}^2 + \text{Var}. \hspace{1cm} (3.41)$$

Combining the unbiased estimators (3.37) and (3.38) with (3.41), we obtain the estimator  

$$|\hat{\text{Bias}}| = \hat{\text{MSE}}_Q - \hat{\text{Var}}_Q \hspace{1cm} (3.42)$$

for $|\text{Bias}|$ of the truncated sample mean $\bar{Z}_q(k, d^*_q)$.

In estimating the performance of MSER-5Y using Equations (3.36) through (3.42), we calculated all statistics over $Q = 1,000$ independent replications of each test process. However in estimating the performance of MSER-5 using Equations (3.36) through (3.42), we only used the $Q$ replications of each
test process for which MSER-5 successfully delivered a truncation point and a truncated sample mean; thus $Q < 1,000$ when MSER-5 was applied to some test processes.

In addition to these statistics, we tabulated the number of failures that occurred in the application of MSER-5 to each test process. Obviously, no such statistic is provided for MSER-5Y. Furthermore, as a result of the failures (or inconclusive runs), the performance measures for MSER-5 are conditional given a successful result in implementing the procedure; therefore, the reported performance measures are averaged only over the number of replications $Q$ where MSER-5 successfully delivered point and CI estimators of the steady-state mean $\mu$. We have also reported an additional statistic for MSER-5 called “Unconditional CI coverage,” which is the percentage of all 1,000 replications on which MSER-5 delivered a CI that covers the true steady-state mean $\mu$. This statistic is intended to characterize the performance of the CI that can be expected on a single application of MSER-5 in practice.
Chapter 4

Results

The purpose of this chapter is to summarize the performance of the two truncation methods MSER-5 and MSER-5Y for the experiments defined in the previous chapter. A list of the statistics collected for each algorithm and their descriptions are provided in Section 3.3.

Tables 4.1 to 4.5 demonstrate the performance of the MSER-5 algorithm and the modified algorithm, MSER-5Y, on 1,000 independent replications of the experiments described in Sections 3.2.1 to 3.2.5, respectively. A brief discussion of the results for each of the experiments will follow. Note that in all the following tables, the estimated standard error of each CI coverage is enclosed in parentheses and given immediately below the associated CI coverage probability.
Table 4.1: Performance of MSER-5 and MSER-5Y in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition

<table>
<thead>
<tr>
<th></th>
<th>Results for MSER-5 Algorithm</th>
<th></th>
<th>Overall Sample Size N</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 – α</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
</tr>
<tr>
<td></td>
<td>Unconditional CI coverage 39.2%</td>
<td>49.5%</td>
<td>67.7%</td>
<td>79.8%</td>
</tr>
<tr>
<td></td>
<td>Conditional CI coverage 61.2%</td>
<td>66.7%</td>
<td>79.7%</td>
<td>86.7%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec. 22.33%</td>
<td>18.45%</td>
<td>13.64%</td>
<td>7.67%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length 1.8323</td>
<td>1.5443</td>
<td>1.1956</td>
<td>0.6871</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length 0.6031</td>
<td>0.3131</td>
<td>0.1312</td>
<td>0.0295</td>
</tr>
<tr>
<td></td>
<td>Conditional CI Coverage 43.7%</td>
<td>54.9%</td>
<td>73.3%</td>
<td>84.9%</td>
</tr>
<tr>
<td></td>
<td>CI Coverage 68.3%</td>
<td>74.0%</td>
<td>86.3%</td>
<td>92.3%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec. 27.03%</td>
<td>22.33%</td>
<td>16.52%</td>
<td>9.28%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length 2.2178</td>
<td>1.8692</td>
<td>1.4472</td>
<td>0.8317</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length 0.8837</td>
<td>0.4588</td>
<td>0.1923</td>
<td>0.0432</td>
</tr>
<tr>
<td></td>
<td>Empirical Point-Estimator Trunc. Sample Mean 8.2058</td>
<td>8.3720</td>
<td>8.7625</td>
<td>8.9622</td>
</tr>
<tr>
<td></td>
<td>MSE 3.4958</td>
<td>1.8867</td>
<td>0.6891</td>
<td>0.1836</td>
</tr>
<tr>
<td></td>
<td>Variance 2.8650</td>
<td>1.4923</td>
<td>0.6327</td>
<td>0.1822</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td># of Failures in 1,000 Runs 360</td>
<td>258</td>
<td>151</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Results for MSER-5Y Algorithm</th>
<th></th>
<th>Overall Sample Size N</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 – α</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
</tr>
<tr>
<td></td>
<td>CI coverage 65.7%</td>
<td>65.4%</td>
<td>79.0%</td>
<td>84.9%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec. 24.95%</td>
<td>19.30%</td>
<td>13.37%</td>
<td>7.30%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length 2.0860</td>
<td>1.6244</td>
<td>1.1754</td>
<td>0.6541</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length 1.2886</td>
<td>0.6532</td>
<td>0.1553</td>
<td>0.0206</td>
</tr>
<tr>
<td></td>
<td>CI Coverage 71.70%</td>
<td>73.60%</td>
<td>85.90%</td>
<td>90.30%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec. 30.26%</td>
<td>23.28%</td>
<td>16.04%</td>
<td>8.72%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length 2.5297</td>
<td>1.9594</td>
<td>1.4099</td>
<td>0.7817</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length 1.9551</td>
<td>0.9794</td>
<td>0.2280</td>
<td>0.0301</td>
</tr>
<tr>
<td></td>
<td>Empirical Point-Estimator Trunc. Sample Mean 8.3599</td>
<td>8.4178</td>
<td>8.7925</td>
<td>8.9640</td>
</tr>
<tr>
<td></td>
<td>MSE 3.4342</td>
<td>2.0662</td>
<td>0.7088</td>
<td>0.1845</td>
</tr>
<tr>
<td></td>
<td>Variance 3.0245</td>
<td>1.7272</td>
<td>0.6657</td>
<td>0.1832</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td># of Failures in 1,000 Runs 360</td>
<td>258</td>
<td>151</td>
<td>80</td>
</tr>
</tbody>
</table>
Table 4.2: Performance of MSER-5 and MSER-5Y in the M/M/1 queue-waiting-time process with 90% server utilization and 113 initial customers

<table>
<thead>
<tr>
<th>Results for MSER-5 Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
</tr>
<tr>
<td>1 - α</td>
</tr>
<tr>
<td>Unconditional CI coverage</td>
</tr>
<tr>
<td>Conditional CI coverage</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
</tr>
<tr>
<td>Var. CI half-length</td>
</tr>
<tr>
<td>Unconditional CI Coverage</td>
</tr>
<tr>
<td>CI Coverage</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
</tr>
<tr>
<td>Var. CI half-length</td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
</tr>
<tr>
<td>Performance Measures</td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>Variance</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td># of Failures in 1,000 Runs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results for MSER-5Y Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
</tr>
<tr>
<td>1 - α</td>
</tr>
<tr>
<td>CI coverage</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
</tr>
<tr>
<td>Var. CI half-length</td>
</tr>
<tr>
<td>CI Coverage</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
</tr>
<tr>
<td>Var. CI half-length</td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
</tr>
<tr>
<td>Performance Measures</td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>Variance</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Table 4.3: Performance of MSER-5 and MSER-5Y in the $M/$M/1/LIFO queue-waiting-time process with 90% server utilization and empty-and-idle initial condition

<table>
<thead>
<tr>
<th>Results for MSER-5 Algorithm</th>
<th>Overall Sample Size $N$</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td>Empirical Perf. Meas.</td>
<td>Unconditional CI coverage</td>
<td>42.7%</td>
<td>55.3%</td>
<td>67.0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conditional CI coverage</td>
<td>77.1%</td>
<td>80.1%</td>
<td>85.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. rel. prec.</td>
<td>15.59%</td>
<td>11.90%</td>
<td>8.04%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. CI half-length</td>
<td>0.5924</td>
<td>0.4624</td>
<td>0.3181</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Var. CI half-length</td>
<td>0.0393</td>
<td>0.0187</td>
<td>0.0061</td>
</tr>
<tr>
<td><strong>90%</strong></td>
<td>CI Coverage</td>
<td>47.0%</td>
<td>59.7%</td>
<td>70.8%</td>
<td>82.2%</td>
</tr>
<tr>
<td></td>
<td>Conditional CI coverage</td>
<td>84.8%</td>
<td>86.5%</td>
<td>90.2%</td>
<td>92.1%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec.</td>
<td>18.87%</td>
<td>14.41%</td>
<td>9.73%</td>
<td>5.03%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length</td>
<td>0.7170</td>
<td>0.5597</td>
<td>0.3850</td>
<td>0.2006</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length</td>
<td>0.0575</td>
<td>0.0274</td>
<td>0.0090</td>
<td>0.0015</td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td>Performance Measures</td>
<td>Overall Sample Size $N$</td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
</tr>
<tr>
<td><strong>Trunc. Sample Mean</strong></td>
<td></td>
<td>3.7996</td>
<td>3.8846</td>
<td>3.9558</td>
<td>3.9905</td>
</tr>
<tr>
<td><strong>MSE</strong></td>
<td></td>
<td>0.1985</td>
<td>0.1033</td>
<td>0.0413</td>
<td>0.0125</td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td></td>
<td>0.1584</td>
<td>0.0900</td>
<td>0.0394</td>
<td>0.0106</td>
</tr>
<tr>
<td><strong>[Bias]</strong></td>
<td></td>
<td>0.2004</td>
<td>0.1154</td>
<td>0.0442</td>
<td>0.0442</td>
</tr>
<tr>
<td><strong># of Failures in 1,000 Runs</strong></td>
<td></td>
<td>446</td>
<td>310</td>
<td>215</td>
<td>108</td>
</tr>
</tbody>
</table>

Results for MSER-5Y Algorithm

<table>
<thead>
<tr>
<th>Confidence-Interval Properties</th>
<th>Overall Sample Size $N$</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td>Empirical Perf. Meas.</td>
<td>CI coverage</td>
<td>76.9%</td>
<td>79.0%</td>
<td>85.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. rel. prec.</td>
<td>15.34%</td>
<td>11.54%</td>
<td>7.69%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. CI half-length</td>
<td>0.5867</td>
<td>0.4506</td>
<td>0.3047</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Var. CI half-length</td>
<td>0.0387</td>
<td>0.0181</td>
<td>0.0042</td>
</tr>
<tr>
<td><strong>90%</strong></td>
<td>CI Coverage</td>
<td>82.1%</td>
<td>86.4%</td>
<td>89.3%</td>
<td>92.8%</td>
</tr>
<tr>
<td></td>
<td>Conditional CI coverage</td>
<td>82.0%</td>
<td>86.4%</td>
<td>89.3%</td>
<td>92.8%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec.</td>
<td>18.36%</td>
<td>13.79%</td>
<td>9.18%</td>
<td>4.75%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length</td>
<td>0.702</td>
<td>0.5384</td>
<td>0.3637</td>
<td>0.1894</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length</td>
<td>0.0563</td>
<td>0.0262</td>
<td>0.0061</td>
<td>0.0004</td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td>Performance Measures</td>
<td>Overall Sample Size $N$</td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
</tr>
<tr>
<td><strong>Trunc. Sample Mean</strong></td>
<td></td>
<td>3.8236</td>
<td>3.9040</td>
<td>3.9599</td>
<td>3.9898</td>
</tr>
<tr>
<td><strong>MSE</strong></td>
<td></td>
<td>0.1900</td>
<td>0.1038</td>
<td>0.0408</td>
<td>0.0119</td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td></td>
<td>0.1669</td>
<td>0.0946</td>
<td>0.0392</td>
<td>0.0103</td>
</tr>
<tr>
<td><strong>[Bias]</strong></td>
<td></td>
<td>0.1764</td>
<td>0.0960</td>
<td>0.0401</td>
<td>0.0401</td>
</tr>
</tbody>
</table>

48
Table 4.4: Performance of MSER-5 and MSER-5Y in the AR(1) process with $\rho=0.995$, $X_0=0$ and $\mu=100$

<table>
<thead>
<tr>
<th>Results for MSER-5 Algorithm</th>
<th>Overall Sample Size $N$</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td><strong>Empirical Perf. Meas.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1-\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unconditional CI coverage</td>
<td></td>
<td>48.6%</td>
<td>60.6%</td>
<td>73.0%</td>
<td>81.9%</td>
</tr>
<tr>
<td>(1.6%)</td>
<td></td>
<td>(1.5%)</td>
<td>(1.4%)</td>
<td>(1.2%)</td>
<td></td>
</tr>
<tr>
<td>Conditional CI coverage</td>
<td></td>
<td>78.8%</td>
<td>82.7%</td>
<td>86.8%</td>
<td>89.1%</td>
</tr>
<tr>
<td>(2.0%)</td>
<td></td>
<td>(1.8%)</td>
<td>(1.5%)</td>
<td>(1.3%)</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td>2.61%</td>
<td>2.12%</td>
<td>1.45%</td>
<td>0.76%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td>2.6037</td>
<td>2.1148</td>
<td>1.4506</td>
<td>0.7564</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td>0.2093</td>
<td>0.1303</td>
<td>0.0573</td>
<td>0.0137</td>
</tr>
<tr>
<td><strong>90%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unconditional CI Coverage</td>
<td></td>
<td>52.8%</td>
<td>67.2%</td>
<td>77.9%</td>
<td>86.8%</td>
</tr>
<tr>
<td>(1.6%)</td>
<td></td>
<td>(1.5%)</td>
<td>(1.3%)</td>
<td>(1.1%)</td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td>85.6%</td>
<td>91.7%</td>
<td>92.6%</td>
<td>94.4%</td>
</tr>
<tr>
<td>(2.0%)</td>
<td></td>
<td>(1.7%)</td>
<td>(1.4%)</td>
<td>(1.1%)</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td>3.15%</td>
<td>2.56%</td>
<td>1.76%</td>
<td>0.92%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td>3.1517</td>
<td>2.5598</td>
<td>1.7559</td>
<td>0.9156</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td>0.3067</td>
<td>0.1909</td>
<td>0.0843</td>
<td>0.0201</td>
</tr>
<tr>
<td><strong>95%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size $N$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td>4.3260</td>
<td>2.1761</td>
<td>0.8660</td>
<td>0.2104</td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td>4.3220</td>
<td>2.1760</td>
<td>0.8640</td>
<td>0.2104</td>
</tr>
<tr>
<td>[Bias]</td>
<td></td>
<td>0.0636</td>
<td>0.0118</td>
<td>0.0433</td>
<td>0.0004</td>
</tr>
<tr>
<td># of Failures in 1,000 Runs</td>
<td></td>
<td>383</td>
<td>267</td>
<td>159</td>
<td>81</td>
</tr>
</tbody>
</table>

Results for MSER-5Y Algorithm

<table>
<thead>
<tr>
<th>Results for MSER-5Y Algorithm</th>
<th>Overall Sample Size $N$</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td><strong>Empirical Perf. Meas.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1-\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI coverage</td>
<td></td>
<td>78.1%</td>
<td>84.8%</td>
<td>86.6%</td>
<td>87.9%</td>
</tr>
<tr>
<td>(1.3%)</td>
<td></td>
<td>(1.1%)</td>
<td>(1.1%)</td>
<td>(1.0%)</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td>2.94%</td>
<td>2.17%</td>
<td>1.42%</td>
<td>0.72%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td>2.9356</td>
<td>1.1737</td>
<td>1.4234</td>
<td>0.7192</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td>0.5598</td>
<td>0.2451</td>
<td>0.0735</td>
<td>0.0079</td>
</tr>
<tr>
<td><strong>90%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td>85.5%</td>
<td>91.6%</td>
<td>90.8%</td>
<td>93.6%</td>
</tr>
<tr>
<td>(1.1%)</td>
<td></td>
<td>(0.9%)</td>
<td>(0.9%)</td>
<td>(0.8%)</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td>3.58%</td>
<td>2.63%</td>
<td>1.71%</td>
<td>0.86%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td>3.5808</td>
<td>2.6306</td>
<td>0.7114</td>
<td>0.8599</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td>0.8783</td>
<td>0.3831</td>
<td>0.1142</td>
<td>0.0120</td>
</tr>
<tr>
<td><strong>95%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size $N$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td>4.8777</td>
<td>2.1000</td>
<td>0.8803</td>
<td>0.2080</td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td>4.8695</td>
<td>2.0987</td>
<td>0.8797</td>
<td>0.2080</td>
</tr>
<tr>
<td>[Bias]</td>
<td></td>
<td>0.0907</td>
<td>0.0362</td>
<td>0.0237</td>
<td>0.0018</td>
</tr>
</tbody>
</table>
Table 4.5: Performance of MSER-5 and MSER-5Y in the ARTOP process

## Results for MSER-5 Algorithm

<table>
<thead>
<tr>
<th>Confidence-Interval Properties</th>
<th>Empirical Perf. Meas.</th>
<th>Overall Sample Size</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
<td>200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unconditional CI coverage</td>
<td>9.0%</td>
<td>12.9%</td>
<td>19.2%</td>
<td>35.9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.9%)</td>
<td>(1.1%)</td>
<td>(1.2%)</td>
<td>(1.5%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conditional CI coverage</td>
<td>41.3%</td>
<td>54.9%</td>
<td>62.3%</td>
<td>75.7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3.0%)</td>
<td>(3.2%)</td>
<td>(2.8%)</td>
<td>(2.0%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>13.54%</td>
<td>12.24%</td>
<td>9.30%</td>
<td>5.80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.2276</td>
<td>0.2178</td>
<td>0.1688</td>
<td>0.1085</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0179</td>
<td>0.0099</td>
<td>0.0036</td>
<td>0.00076</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unconditional CI Coverage</td>
<td>10.6%</td>
<td>14.8%</td>
<td>21.2%</td>
<td>39.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.0%)</td>
<td>(1.1%)</td>
<td>(1.3%)</td>
<td>(1.5%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td>48.6%</td>
<td>63.0%</td>
<td>68.8%</td>
<td>83.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3.4%)</td>
<td>(3.1%)</td>
<td>(2.6%)</td>
<td>(1.7%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>16.39%</td>
<td>14.81%</td>
<td>11.26%</td>
<td>7.02%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.2755</td>
<td>0.2636</td>
<td>0.2043</td>
<td>0.1313</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0262</td>
<td>0.0145</td>
<td>0.0053</td>
<td>0.0011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td>60.7%</td>
<td>73.1%</td>
<td>76.0%</td>
<td>85.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.5%)</td>
<td>(1.4%)</td>
<td>(1.3%)</td>
<td>(1.1%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>21.52%</td>
<td>17.93%</td>
<td>12.70%</td>
<td>7.27%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.3817</td>
<td>0.3290</td>
<td>0.2356</td>
<td>0.1368</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0869</td>
<td>0.0422</td>
<td>0.0205</td>
<td>0.0026</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
<td>200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>1.6804</td>
<td>1.7798</td>
<td>1.8143</td>
<td>1.8705</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.1126</td>
<td>0.1460</td>
<td>0.0279</td>
<td>0.0067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>0.0603</td>
<td>0.1293</td>
<td>0.0189</td>
<td>0.0052</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.2286</td>
<td>0.1292</td>
<td>0.0947</td>
<td>0.0385</td>
<td></td>
</tr>
<tr>
<td># of Failures in 1,000 Runs</td>
<td>782</td>
<td>765</td>
<td>692</td>
<td>526</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Results for MSER-5Y Algorithm

<table>
<thead>
<tr>
<th>Confidence-Interval Properties</th>
<th>Empirical Perf. Meas.</th>
<th>Overall Sample Size</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
<td>200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI coverage</td>
<td>54.6%</td>
<td>66.0%</td>
<td>69.2%</td>
<td>79.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.6%)</td>
<td>(1.5%)</td>
<td>(1.5%)</td>
<td>(1.3%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>17.73%</td>
<td>14.86%</td>
<td>10.60%</td>
<td>6.09%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.3145</td>
<td>0.2728</td>
<td>0.1966</td>
<td>0.1146</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0593</td>
<td>0.0290</td>
<td>0.0143</td>
<td>0.0018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td>60.7%</td>
<td>73.1%</td>
<td>76.0%</td>
<td>85.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.5%)</td>
<td>(1.4%)</td>
<td>(1.3%)</td>
<td>(1.1%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>21.52%</td>
<td>17.93%</td>
<td>12.70%</td>
<td>7.27%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.3817</td>
<td>0.3290</td>
<td>0.2356</td>
<td>0.1368</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0869</td>
<td>0.0422</td>
<td>0.0205</td>
<td>0.0026</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td>60.7%</td>
<td>73.1%</td>
<td>76.0%</td>
<td>85.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.5%)</td>
<td>(1.4%)</td>
<td>(1.3%)</td>
<td>(1.1%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>21.52%</td>
<td>17.93%</td>
<td>12.70%</td>
<td>7.27%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.3817</td>
<td>0.3290</td>
<td>0.2356</td>
<td>0.1368</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0869</td>
<td>0.0422</td>
<td>0.0205</td>
<td>0.0026</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
<td>200,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>1.7734</td>
<td>1.8352</td>
<td>1.8546</td>
<td>1.8825</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.1207</td>
<td>0.0643</td>
<td>0.0313</td>
<td>0.0073</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>0.1023</td>
<td>0.0589</td>
<td>0.0283</td>
<td>0.0066</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.1356</td>
<td>0.0738</td>
<td>0.0544</td>
<td>0.0265</td>
<td></td>
</tr>
</tbody>
</table>
For the M/M/1 queue-waiting time process with empty-and-idle initial condition and steady-state server utilization $\rho=0.9$, the results in Table 4.1 indicate an overall improvement in the results of MSER-5Y over MSER-5. In terms of the CI coverage probabilities, MSER-5Y yielded better CI coverages than MSER-5. The difference was more significant in smaller sample sizes and as the sample size got larger the two methods deliver closer CI coverages with the MSER-5Y’s CI coverage probability being marginally acceptable at sample size $N=200,000$ (i.e. 85% for 90% nominal coverage probability for $N=200,000$). It is worth mentioning that the empirical CI coverage probabilities for both methods over small and medium sample sizes are still unacceptable as they are significantly below the corresponding user-specified nominal coverage probabilities; nevertheless the coverage probabilities delivered by MSER-5Y are much better and at $N=200,000$ it gets close to the specified coverage value.

In addition to better CI coverage probabilities, MSER-5Y delivered better point estimates of the steady state mean as well. The biases in the point estimates were lower for MSER-5Y over all sample sizes. Again as the sample size got larger, the number of failures in MSER-5 decreased and the point estimates delivered by both methods seemed to converge to the true steady state mean. The MSE in the estimates of the steady-state mean delivered by both methods are very close in value. That also suggests that since bias is slightly lower for MSER-5Y, to have close MSEs, the variance should be slightly higher for MSER-5Y and that was confirmed by the results.

Another point worth mentioning here is the average relative precision that decreased as the sample size increased. At sample size $N=20,000$ where acceptable estimates of the CI is achieved, the average relative precision drops down to less than 10% which confirms the suggestion made by Heidelberger and Lewis (1983) to require at least 10% precision for a “valid” CI estimate.

The final observation is the relatively large number of failures in MSER-5. For example, results in Table 4.1 suggest that MSER-5 fails to deliver any results over 35% of the time for a sample of size $N=10,000$; whereas, MSER-5Y always delivers results which are at least slightly superior and based on those results determines the sample size required to ensure acceptable performance by MSER-5Y.

A comparison of the empirical distribution of the truncated means for both methods is provided for sample sizes of 10,000 and 20,000 in the histograms in Figures 4.1 and 4.2 respectively. The histograms confirm that the additional observations in MSER-5Y from the replications that were not rejected in MSER-5Y on the contrary to MSER-5, improve the empirical distribution as they are more centered around the true mean.
Figure 4.1: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 10,000)

Figure 4.2: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 20,000)
Table 4.2 summarizes the results for the $M/M/1$ queuing system with 113 initial customers and server utilization $\rho=0.9$. The steady-state mean waiting time for this process is $\mu=9.0$. From the results, we concluded that MSER-5Y delivered improved small-sample performance in comparison with MSER-5. With respect to point estimator accuracy, MSER-5Y outperformed MSER-5 by delivering lower bias values for all sample sizes, more significant for smaller sample sizes and as the sample size increased the estimates of the two methods got closer in value. The CI coverage probabilities delivered by MSER-5Y are also better the ones delivered by MSER-5. Again, the CI coverages delivered by both algorithms were lower than the nominal coverage probabilities for sample sizes of $N=10,000$, 20,000 and 50,000 and as the sample size got larger, the empirical CI coverage got closer to the nominal value until for $N=200,000$ the empirical values deviated from the nominal value only slightly and can be considered practically the same.

A comparison of the distribution of the truncated means delivered by the two methods is provided in the following histograms.
Figure 4.3: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with $X_0 = 113$ and $\rho = 0.9$ (Sample size 10,000)

Figure 4.4: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1 queue-waiting-time process for queue with $X_0 = 113$ and $\rho = 0.9$ (Sample size 20,000)
Table 4.3 summarizes the results for the queue waiting time process in the $M/M/1/LIFO$ queueing system. The steady-state mean queue waiting time for this process is $\mu=3.2$. From the table it can be observed that for medium and large sample sizes, MSER-5Y delivered acceptable results in terms of both the CI coverage probabilities and the point estimator. For small sample sizes there is a significant CI undercoverage that indicates the need for larger sample sizes. The point estimator is consistently better in MSER-5Y for all sample sizes and as the sample increased the difference between the point estimates from the two methods got smaller as they got closer to the steady-state value. The failure rate for MSER-5 is still considerably high in this test process. Ranging from 45% to 11% for different sample sizes.
Figure 4.5: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1/LIFO queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 10,000)

Figure 4.6: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to M/M/1/LIFO queue-waiting-time process for queue with $X_0 = 0$ and $\rho = 0.9$ (Sample size 20,000)
Table 4.4 summarizes the results for the AR(1) process with autoregressive parameter $\rho=0.995$. The steady-state mean of this process is determined as $\mu=100$. This process is characterized by a high level of positive correlation between successive observations and a large negative initialization bias which severely distorts the behavior of the classical batch means method. MSER-5Y outperforms MSER-5 in the CI coverage probabilities. The point estimates of MSER-5 were slightly better in some sample sizes but again as the sample size got larger the estimates from both methods delivered close point estimates that are free of initialization bias. An interesting observation in this test process is the low average relative precision reported for the CIs estimated for both MSER-5 and MSER-5Y in the range of 3% to 0.6% even in cases were the empirical CI coverage probability were not acceptable.

The following figures provide a comparison of the empirical distribution of the truncated means delivered by the two methods.
Figure 4.7: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to AR(1) process with $\rho = 0.995$ (Sample size 10,000)

Figure 4.8: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to AR(1) process with $\rho = 0.995$ (Sample size 20,000)
Table 4.5 summarizes the results for the ARTOP process as described in Section 3.2.5. The steady-state mean of this process is $\mu=1.9091$. From the results in Table 4.5, we can see that CI coverage probabilities for MSER-5Y were consistently better in MSER-5Y, even though both MSER-5 and MSER-5Y delivered unacceptable CI coverages. The coverages improved with increased sample sizes, and for $N=200,000$ the CI coverage delivered by MSER-5Y was close to the specified coverage levels. The comparison of the point estimators for the two algorithms indicates superior results for MSER-5Y with lower bias over all sample sizes. Another observation is the very high rate of failure in MSER-5 where the algorithm was unable to deliver any results, whereas MSER-5Y delivers results in those cases and determines the sample size required to ensure acceptable performance.

The following histograms depict the empirical distributions of the truncated means delivered by the two methods for 10,000 and 20,000 sample sizes. The large number of failures in MSER-5 and improvement in the results in MSER-5Y by having more density around the true mean with the “saved” replications is evident in the graphs.
Figure 4.9: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to ARTOP process (Sample size 10,000)

Figure 4.10: Empirical distribution of truncated sample mean for MSER-5 and MSER-5Y when applied to ARTOP process (Sample size 20,000)
Chapter 5

Conclusion

5.1. General Conclusions

The purpose of this research was to introduce MSER-5Y, an improved version of the MSER-5 procedure for handling the start-up problem in steady-state simulation analysis. MSER-5Y is intended to be a fail-safe method that takes advantage of a systematic approach to deliver a valid CI estimator for the steady-state mean as well as a mechanism to adaptively decide about continuing the simulation experiment until the desired estimator accuracy is achieved. To fulfill this purpose, we looked at the research studies in the literature on the MSER-5 truncation heuristic; and based on the shortcomings reported for the MSER-5 algorithm, we developed MSER-5Y. To evaluate the performance of the new method, we automated the two algorithms and applied them to data from five different test processes, with each experiment on each test process consisting of 1,000 replications yielding time series of length $N$, where we let $N = 10,000, 20,000, 50,000$ and $200,000$ to represent small, medium, and large simulation run lengths. The experiments were carefully selected to provide test problems that are more nearly typical of practical applications as well as relatively extreme examples of nonnormal, correlated simulation output processes that in most cases are contaminated with initialization bias. The results of each experiment were analyzed, and the performance of each algorithm was evaluated based on performance measures that capture the ability of the algorithms to deliver valid point estimators of the steady-state mean as well as precise and reliable CI estimators of the steady-state mean. Based on the performance evaluation, the following conclusions can be made:

- MSER-5Y outperformed MSER-5 in almost all the test problems, delivering better point estimators of the steady-state mean as well as greatly improved CI coverage probabilities;

- The improvement in performance achieved by MSER-5Y, in most cases, was more pronounced in small and medium samples. And, as the sample size got larger, the estimators delivered by both methods converged as they got closer to the steady-state mean;
• The key advantage of MSER-5Y over MSER-5 is that it provides a framework for a procedure that sequentially allows the user to decide on continuing the experiment and it provides the user with an estimate of how much more data will be required to deliver estimators of acceptable precision;

• Overall, for large sample sizes we judged the performance of MSER-5Y’s CIs to be marginally acceptable. In all our experiments we have found that, when supplied with a sufficiently large sample, MSER-5Y can deliver acceptable point and CI estimators. For the fixed sample size of $N=200,000$, the empirical coverage probabilities in almost all the test problems differ from their respective nominal levels by less than 5% for both 90% and 95% nominal CI coverage levels. We consider these differences to be practically insignificant;

• Despite the improvements in CI coverage probabilities achieved by the modifications made in MSER-5Y, there still is an undercoverage problem for the small and medium sample sizes, where the empirical coverage probabilities of the CIs delivered by MSER-5Y are significantly below the corresponding user-specified nominal coverage probabilities.

5.2. Contributions

The main contribution of this study was to introduce a modification of the MSER-5 procedure that has a systematic approach to deliver reliable point and CI estimators for the steady-state mean. This approach also provides the user with specific guidance on planning follow-up experiments to achieve a given level of accuracy in simulation-generated statistics.

5.3. Future Research of the Research

Based on this research, the following recommendations are made for future work:

• To develop a fully sequential version of MSER-5Y that requests additional observations in real time (i.e., “on the fly” during the execution of a single simulation run) as needed and then delivers updated point and CI estimators of the steady-state mean after the simulation has delivered the requested additional observations.

• The experimental performance evaluation in the current research should be substantially expanded to include a much greater diversity of test processes with different types of transient behavior.

• To develop the appropriate portable code for MSER-5Y that can be directly invoked in popular simulation software such as Arena for automatic output analysis.
REFERENCES


% Code for implementing the OLD-MSER5 Algorithm

% Basic setting of variables

N1=20000;
numReps = 1000;
alpha1 = 0.10; %100(1-Alpha1) Confidence Interval ---> 90%
alpha2 = 0.05; %100(1-Alpha2) Confidence Interval ---> 95%
fail = 0;
Coverage1 = 0;
Coverage2 = 0;
Nrand = 400000;

% Initialization

N2=floor(N1/5);
%Use this only when Generating tblX/ Change Name according to sample size
%tblX=zeros(N1,numReps);
stat = Inf(1,N2-2);
tblMeanTruncInf=Inf(1,numReps);
tblHL1Inf=Inf(1,numReps);
tblHL2Inf=Inf(1,numReps);
tblRelPre1Inf=Inf(1,numReps);
tblRelPre2Inf=Inf(1,numReps);
failRep=zeros(1,numReps);
tblTrunc1=Inf(1,numReps);
tblTrunc2=Inf(1,numReps);

% Variables particular to each process

%M/M/1 No Customer
TrueMean=9;
Ns=N1;
X0=0;

%M/M/1 113 Customer
% TrueMean=9;
% Ns=N1;
for rep=1:numReps

    disp 'Starting replication: '
    rep

    % Generate/Read sample of size N1

    % M/M/1 No Customer
    x1=rand(1,Nrand);
    x2=rand(1,Nrand);
    X=FuncMM1Q(x1,x2,Ns,X0);
    tblX3(1:N1,rep)=X;
    % Use this only when reading tblX/ Change Name according to sample size
    X=tblX2(1:N1,rep);

    % M/M/1 113 Customer
    x1=rand(1,Nrand);
    x2=rand(1,Nrand);
    X=FuncMM1Q(x1,x2,Ns,X0);
    tblX1131(1:N1,rep)=X;
    % Use this only when reading tblX/ Change Name according to sample size
\% \texttt{X=tblX1132(1:N1,rep);}

\% M/M/1 LIFO
\% \texttt{X=tblXLIFO1(1:N1,rep);}

\% AR(1)
\% \texttt{X=FuncAR1(Mu,Rho,Ns,X0);}
\% tblXAR12(1:N1,rep)=X;
\% Use this only when reading tblX/ Change Name according to sample size
\% \texttt{X=tblXAR11(1:N1,rep);}

\% ARTOP
\% \texttt{X=FuncARTOP(Xi,Psi,Rho,Ns,Z0);}
\% tblXARTOP1(1:N1,rep)=X;
\% Use this only when reading tblX/ Change Name according to sample size
\% \texttt{X=tblXARTOP2(1:N1,rep);}

\% Batch data into batches of size 5
\% for i=N1:-5:1
\% \hspace{1cm} Z(i/5)=mean(X((i-4):i));
\% end

\% Calculate statistic (Half Length) for truncation points 0 to N2-2 and Find the
\% truncation point \( d \) that minimizes the statistic
\% statistic=1000;
\% for j=1:(N2-2) \% Truncation point \( d=j-1 \)
\% \hspace{1cm} stat(j)=var(Z((j+1):N2),1)/(N2-j+1);
\% \hspace{1cm} tblStat(j,rep)=stat(j);
\% if stat(j)<=statistic
\% \hspace{1cm} statistic=stat(j);
\% \hspace{1cm} trunc2=j-1;
\% \hspace{1cm} trunc1=((j-1)+5);
\% end
\% end

\% Fails if truncation point \( \geq \) half the data
\% if trunc2>=floor(N2/2)
\% \hspace{1cm} fail=fail+1;
\% \hspace{1cm} failRep(rep)=rep;
% Continue if truncation point< half the data

else %if trunc2<floor(N2/2)

% Batch into 20 batches
bcount=20;
bsize=floor((N2-trunc2)/bcount);

W=zeros(1,bcount);
for j=1:bcount
    W(j)=mean(Z(((j-1)*bsize+1+trunc2):(j*bsize+trunc2)));% Batch into 20 batches
end

% Calculate statistics
MeanW=mean(W);
VarW=var(W);

meanfinal=MeanW;

%Confidence Level 90%, tinv(0.95,19)
HL1=tinv(1-alpha1/2,bcount-1)*sqrt(VarW/bcount);

%Confidence Level 95%, tinv(0.975,19)
HL2=tinv(1-alpha2/2,bcount-1)*sqrt(VarW/bcount);

if TrueMean>=meanfinal-HL1 && TrueMean<=meanfinal+HL1
    Coverage1=Coverage1+1;
end

if TrueMean>=meanfinal-HL2 && TrueMean<=meanfinal+HL2
    Coverage2=Coverage2+1;
end

% Record statistics for each rep in an array

% tblMeanTruncInf(rep)=meanfinal;
tblHL1Inf(rep)=HL1;
tblHL2Inf(rep)=HL2;
tblRelPreInf(rep)=HL1/meanfinal;
tblRelPre2Inf(rep)=HL2/meanfinal;
tblTrunc1(rep)=trunc1;
tblTrunc2(rep)=trunc2;

end
end

format compact

tblfailRep=failRep (failRep >0);
tblTrunc1Final = tblTrunc1 (tblTrunc1 < floor(N1/2));

% Remove values in the arrays caused by initialization
%__________________________________________________________

tblMeanTrunc = tblMeanTruncInf (tblMeanTruncInf ~= Inf);
tblHL1 = tblHL1Inf (tblHL1Inf ~= Inf);
tblHL2 = tblHL2Inf (tblHL2Inf ~= Inf);
tblRelPre1 = tblRelPre1Inf (tblRelPre1Inf ~= Inf);
tblRelPre2 = tblRelPre2Inf (tblRelPre2Inf ~= Inf);

UnconditionalCoverage1 =(Coverage1 / numReps)*100
CICoverage1 = (Coverage1 / (numReps - fail))*100
AvgRelPre1 = mean(tblRelPre1 (1: numReps - fail))
AvgHL1 = mean(tblHL1 (1: numReps - fail))
VarHL1 = var(tblHL1 (1: numReps - fail))

UnconditionalCoverage2 = (Coverage2 / numReps)*100
CICoverage2 = (Coverage2 / (numReps - fail))*100
AvgRelPre2 = mean(tblRelPre2 (1: numReps - fail))
AvgHL2 = mean(tblHL2 (1: numReps - fail))
VarHL2 = var(tblHL2 (1: numReps - fail))

GrandMean = mean(tblMeanTrunc (1: numReps - fail))
Variance = var(tblMeanTrunc (1: numReps - fail))
Bias = abs(TrueMean - GrandMean)
MSE = Variance + (Bias)^2
fail
A2. Matlab Code for MSER-5Y algorithm

% Code for implementing the OLD-MSER5 Algorithm

% Basic setting of variables

N1=10000;
numReps=1000;
alp1a=0.10; %100(1−Alpha1) Confidence Interval —> 90%
alp1a2=0.05; %100(1−Alpha2) Confidence Interval —> 95%
fail=0;
Coverage1=0;
Coverage2=0;
Nrand=400000;

% Initialization

N2=floor(N1/5);
%sUse this only when Generating tblX/ Change Name according to sample size
%tblX=zeros(N1,numReps);
stat=Inf(1,N2-2);
tblMeanTruncInf=Inf(1,numReps);
tblHL1Inf=Inf(1,numReps);
tblHL2Inf=Inf(1,numReps);
tblRelPre1Inf=Inf(1,numReps);
tblRelPre2Inf=Inf(1,numReps);
failRep=zeros(1,numReps);
tblTrunc1=Inf(1,numReps);
tblTrunc2=Inf(1,numReps);

% Variables particular to each process

%M/M/1 No Customer
%TrueMean=9;
%Ns=N1;
%X0=0;

%M/M/1 113 Customer
TrueMean=9;
Ns=N1;
X0=113;

% M/M/1 LIFO
% TrueMean=4;
% tblXLIFO1Col=csvread('mm1lifowaits1.dat');
% tblXLIFO1=reshape(tblXLIFO1Col,10000,1000);

% AR(1)
% TrueMean=100;
% Ns=N1;
% X0=0;
% Mu=100;
% Rho=0.995;

% ARTOP
% TrueMean=1.9091;
% Ns=N1;
% Z0=3.4;
% Xi=1;
% Psi=2.1;
% Rho=0.995;

for rep=1:numReps

disp('Starting replication:');
rep

% Generate/Read sample of size N1

% M/M/1 No Customer
% x1=rand(1,Nrand);
% x2=rand(1,Nrand);
% X=FuncMM1Q(x1,x2,Ns,X0);
% tblX3(1:N1,rep)=X;
% Use this only when reading tblX/ Change Name according to sample size
% X=tblX2(1:N1,rep);

% M/M/1 113 Customer
% x1=rand(1,Nrand);
% x2=rand(1,Nrand);
% X=FuncMM1Q(x1,x2,Ns,X0);
% tblX1131(1:N1,rep)=X;
% Use this only when reading tblX/ Change Name according to sample size
X=tblX1131(1:N1,rep);

% M/M/1 LIFO
X=tblXLIFO1(1:N1,rep);

% AR(1)
X=FuncAR1(Mu,Rho,Ns,X0);
% tblXAR12(1:N1,rep)=X;
% Use this only when reading tblX/ Change Name according to sample size
% X=tblXAR11(1:N1,rep);

% ARTOP
X=FuncARTOP(Xi,Psi,Rho,Ns,Z0);
% tblXARTOP1(1:N1,rep)=X;
% Use this only when reading tblX/ Change Name according to sample size
% X=tblXARTOP2(1:N1,rep);

% Batch data into batches of size 5
for i=N1:-5:1
    Z(i/5)=mean(X((i-4):i));
end

% Calculate statistic (Half Length) for truncation points 0 to N2−2 and Find the
% truncation point d that minimizes the statistic
statistic=1000;
for j=1:(N2−2) %Truncation point d=j−1
    stat(j)=var(Z((j+1):N2),1)/(N2−j+1);
    tblStat(j,rep)=stat(j);
    if stat(j)<=statistic
        statistic=stat(j);
        trunc2=j−1;
        trunc1=((j−1)*5);
    end
end

% Fails if truncation point >= half the data
if trunc2>=floor(N2/2)
    fail=fail+1;
end
failRep(rep)=rep;

% Continue if truncation point < half the data
else %if trunc2<floor(N2/2)
    % Batch into 20 batches
    bcount=20;
    bsize=floor ((N2-trunc2)/bcount);
    W=zeros(1,bcount);
    for j=1:bcount
        W(j)=mean(Z((j-1)*bsize+1+trunc2):(j*bsize+trunc2));
    end

% Calculate statistics
MeanW=mean(W);
VarW=var(W):

meanfinal=MeanW;
%Confidence Level 90%, tinv (0.95,19)
HL1=tinv (1-alpha1/2,bcount-1)*sqrt (VarW/bcount);
%Confidence Level 95%, tinv (0.975,19)
HL2=tinv (1-alpha2/2,bcount-1)*sqrt (VarW/bcount);

if TrueMean>=meanfinal-HL1 && TrueMean<=meanfinal+HL1
    Coverage1=Coverage1+1;
end
if TrueMean>=meanfinal-HL2 && TrueMean<=meanfinal+HL2
    Coverage2=Coverage2+1;
end

% Record statistics for each rep in an array
tblMeanTruncInf(rep)=meanfinal;
tblHL1Inf(rep)=HL1;
tblHL2Inf(rep)=HL2;
tblRelPre1Inf(rep)=HL1/meanfinal;
tblRelPre2Inf(rep)=HL2/meanfinal;
tblTrunc1(rep)=trunc1;
tblTrunc2(rep)=trunc2;
end
end

format compact

tblfailRep=failRep(failRep > 0);
tblTrunc1Final=tblTrunc1(tblTrunc1 < floor(N1/2));

% Remove values in the arrays caused by initialization
 tblMeanTrunc=tblMeanTruncInf(tblMeanTruncInf != Inf);
tblHL1=tblHL1Inf(tblHL1Inf != Inf);
tblHL2=tblHL2Inf(tblHL2Inf != Inf);
tblRelPre1=tblRelPre1Inf(tblRelPre1Inf != Inf);
tblRelPre2=tblRelPre2Inf(tblRelPre2Inf != Inf);

UnconditionalCoverage1=(Coverage1/numReps)*100
CICoverage1=(Coverage1/(numReps-fail))*100
AvgRelPre1=mean(tblRelPre1(1:numReps-fail))
AvgHL1=mean(tblHL1(1:numReps-fail))
VarHL1=var(tblHL1(1:numReps-fail))

UnconditionalCoverage2=(Coverage2/numReps)*100
CICoverage2=(Coverage2/(numReps-fail))*100
AvgRelPre2=mean(tblRelPre2(1:numReps-fail))
AvgHL2=mean(tblHL2(1:numReps-fail))
VarHL2=var(tblHL2(1:numReps-fail))

GrandMean=mean(tblMeanTrunc(1:numReps-fail))
Variance=var(tblMeanTrunc(1:numReps-fail))
Bias=abs(TrueMean-GrandMean)
MSE=Variance+(Bias)^2
fail