DETECTION OF STEADY STATE IN DISCRETE EVENT DYNAMIC SYSTEMS:
AN ANALYSIS OF HEURISTICS

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science Systems Engineering

by

Mary Alice McClamon

August, 1990

APPROVAL SHEET

This thesis is submitted in partial fulfillment of the requirements for the degree of

Master of Science Systems Engineering

Author

This thesis has been read and approved by the examining Committee:

R. Preston White Jr.

Thesis Advisor

Julia Pet-Edwards

Accepted for the School of Engineering and Applied Science:

Edgar A Starke, Jr.

Dean, School of Engineering and
Applied Science

August, 1990

ABSTRACT

The choice of initial conditions is a fundamental issue in the experimental analysis of non-terminating, stochastic, discrete-event dynamic systems (DEDS). Because output observations are sequentially correlated, cumulative performance estimators are biased when initial observations are not representative of steady-state operation. A poor choice of initial conditions, at best, requires a larger sample of steady-state operations so as to dilute the initialization bias. At worst, the bias goes undetected, and the result is that insufficient data are collected to ensure accurate statistical performance estimates.

Initialization bias is most often associated with discrete-event simulation, in which initial conditions are generally selected to convenience the analyst. In the simulation literature, this issue is called the "initial transient" or "start-up" problem. While various solutions to this problem have been proposed, by far the most common are those based on truncation of the output sequence. An observation in the sequence is identified as the first point representing steady-state operation. All prior observations are then deleted and the truncated sequence is retained for output analysis.

This research contributes a new and superior method for quantifying the performance of steady-state detection heuistics used to determine truncation points. Prior studies have concluded that existing heuristics are either ineffective, because the implied number of observations truncated is too few, or inefficient, because the implied number of observations truncated is too great. These prior studies are reviewed and several methodological deficiencies are identified. The method developed and applied here corrects these deficiencies by posing a more appropriate set of evaluation criteria and a more representative set of benchmark models.

This research also contributes a new and superior steady-state detection heuristic. The Confidence Maximization Rule (CMR) compares sample confidence intervals across candidate truncation points and identifies the onset of steady state as the point that minimizes the halfwidth of the truncated sample. The CMR was suggested by and tested using the method previously developed.

The CMR was evaluated against two existing heuristics that do not require pilot runs. The implied truncation point was computed using each heuristic for each of ten fixed-length runs of five different DEDS benchmark models. In most cases, the performance of the CMR was superior in terms of coverage and consistency. In all cases, the performance of the CMR was at least as good as the next best alternative. For the benchmark problems with significant initialization bias, the CMR also was shown to yield estimates comparable to those derived from untruncated samples with six times the number of observations.

TABLE OF CONTENTS

CHAPTER PAGE

APPENDIX PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

TABLE PAGE

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Output analysis of stochastic discrete-event dynamic systems (DEDS) is the process of estimating the population statistics of selected system response variables from sample data generated during simulation experiments. Reliable methods for DEDS output analysis are well established, and new and potentially more efficient methods are evolving rapidly. For non-terminating DEDS, however, these methods invariably presuppose that the available sample data are representative of steady-state system operation, unbiased by transients.

Techniques required to remove the transient bias from non-terminating simulation data (caused by arbitrary initial conditions) have received comparatively less attention. A range of heuristics has been developed to address the so-called simulation start-up problem, but all of these appear to be inadequate in one way or another. The root of this difficulty can be traced, at least in part, to the lack of a clear and consistent *operational* definition of steady state for DEDS that can be used in creating and judging such heuristics.

The purposes of this research are:

1) to explore the fundamental issues that make the concept of steady state difficult to interpret for DEDS experiments,

2) to develop operational definitions of steady state and system settling time which are appropriate for DEDS simulation studies,

3) to develop a procedure to assess the performance of settling-time heuristics, in the light of these issues and operational definitions,

4) to reassess the performance of several settling-time heuristics by applying this procedure, and

5) to develop and test a new settling-time heuristic with potentially improved performance.

In this chapter, we first consider the objectives of output analysis, as determined by the purpose of DEDS simulation studies. Terminating and non-terminating systems are defined, along with the problems of achieving analysis objectives for each. We review several informal and formal definitions of steady state, taken from the simulation literature, and consider what would be required to use these definitions in the course of a simulation study. An alternative definition of DEDS steady state and settling time, based solely on sample statistics, is then proposed. A brief overview of the remainder of the thesis is presented in the final section.

1.2 TERMINATING AND NON-TERMINATING SYSTEMS

All simulation studies have a directed purpose. The information requirements of the decisionmaker determine this purpose and govern every aspect of the study. With respect to output analysis, the purpose of the simulation study determines the system variables to be observed and the precision and confidence required for these observations.

Fulfillment of these requirements calls for the resolution of two fundamental operational issues. First, the length of each simulation run or replication must be determined, in terms of the simulated time or the number of observations per replication. Second, the number of required replications must be established. These two issues relate the variability in the simulation output data to the confidence levels that can be achieved and determine how well the purpose of the simulation can be fulfilled within the time and budget allowed.

Determination of the appropriate replication length and number of replications depends mainly on whether the system being simulated is *terminating* or *non-terminating*. Each of these simulation types has its own methodologies and techniques for output analysis. A terminating simulation is one for which natural initial and terminating conditions can be easily defined. An example is the simulation of a bank or store that opens and closes periodically at regular time intervals. The analysis of such a simulation concerns techniques for output analysis. A terminating simulation is one for which natural initial and terminating conditions can be easily defined. An example is the simulation of a bank or store that opens and closes periodically at regular time intervals. The analysis of such a simulation concerns the behavior of the system over the *entire* time interval defined by these conditions. Initial conditions may have a significant effect on this behavior and are of interest in the output studies.

A non-terminating simulation, on the other hand, has no natural initial or terminal conditions. An example is the simulation of a factory or telecommunications network that operates twenty-four hours a day, seven days a week. The objective of output analysis for such a simulation is to study the behavior of the system over a representative period of operation, after the initial conditions of the simulation no longer affect its behavior.

Replication length and number of replications are decided differently for terminating and non-terminating simulations. For terminating simulations, both questions have natural answers. Replication length is set by the actual initial and terminal conditions of the simulated system. The required number of replications is determined by the variability of the output data, just as the variability of the data across periods of operation determines the number of periods the actual system would have to be observed experimentally. In general, observations across replications are statistically independent for simulations using independent random number streams; therefore, standard statistical techniques can be used to determine the number of replications required.

For non-terminating simulations, the resolution of these operational issues is less straightforward. Because initial and terminal conditions are not naturally defined, these must be invented for the analysis. The analysis of this data requires some means of defining artificial initial and terminal conditions, in order to block observations into finite, statistically independent sets. These filtered sets can then be

The need for such rules is not unique to simulation output analysis, but is the same for any empirical study of a non-terminating stochastic process, regardless of whether experimentation is conducted on the actual system or on a model of the system. The simple truth that constrains both situations is that observations cannot be made infinitely into the future, so the analysis must rely on sample data and must draw inferences based on these data. Even when experimenting on the actual system, these compromises must be made. A sample drawn from the actual system will not necessarily reflect the average, long-term behavior of the system; in fact, such behavior may not even exist. The inherent difficulty of sample data constraints is simply exaggerated in simulation studies because the simulationist has greater control over data collection and analysis decisions.

1.3 DEFINITIONS OF STEADY STATE

The concept of *steady-state* behavior is essential to the analysis of non-terminating simulations. Formal definitions of steady state provide a theoretical basis for developing rules used to filter sequences of sample data. This section introduces the concept of steady state and discusses its treatment in the simulation literature.

The purpose of a simulation study determines the system operating condition to be studied; that is, transient or steady-state operation. For naturally non-terminating systems, long-term operating characteristics are typically the most significant. The usual assumption is that these systems are not chaotic. That is, for stable, non-terminating systems, the simulation output eventually will settle into some steady-state pattern, from which the simulationist can derive these long-term characteristics. As Law and Kelton (1982) observe, "Many books and papers on simulation make a statement like 'It is desired to estimate some measure of performance for a system that is operating in steady state.'" Similarly, Pegden (1985) says of non-terminating simulations, "...we are interested in defining the *steady state* performance of the system." The point is that, without the assumption of a steady state, long-term stochastic behavior cannot be determined from a finite sampling experiment, and simulation is a doubtful endeavor.

The importance of steady-state *detection* is easy to see. The transient response that results from effects of the arbitrary initial conditions of the simulation run are not of interest in this type of system. If the weight of this initial transient behavior significantly corrupts the steady-state estimates, the statistical results will be biased. The obvious remedy is either to de-emphasize the effects of the transient response on the total system behavior by using large samples, or to eliminate the transient response from the system analysis by truncating the observations prior to steady state. As Emshoff and Sisson (1970) contend, "A good experimental design insures that the results during such a transitional phase are insignificant or are not included in the analysis."

The problem is that this objective is not as easy to achieve as it may seem. A means for recognizing steady-state operation is required, whether we choose to overwhelm any transient bias in a sample with a sufficiently large number of steady-state observations, or to purge the transient bias from these observations. This in turn requires an operational definition of steady state for DEDS.

Control theory provides such an operational definition for deterministic continuous- or discrete-time dynamic systems with damping. A continuous process x(t) is said to achieve a steady state, x_{ss}, when

$$1+\varepsilon \geq \frac{x(t)}{x_{ss}} \geq 1-\varepsilon \quad \text{for all} \quad t \geq ts,$$

where t_s is called the system *settling time* associated with an (ε * 100) percent settling band. Two percent and five percent settling times are common performance measures for the classical analysis and design of control systems. For nonlinear deterministic systems, the value of x_{ss} is a function of the initial condition, x(0). For stable, linear, deterministic systems, a unique steady state exists and both steady state and approximate settling times can be determined analytically.

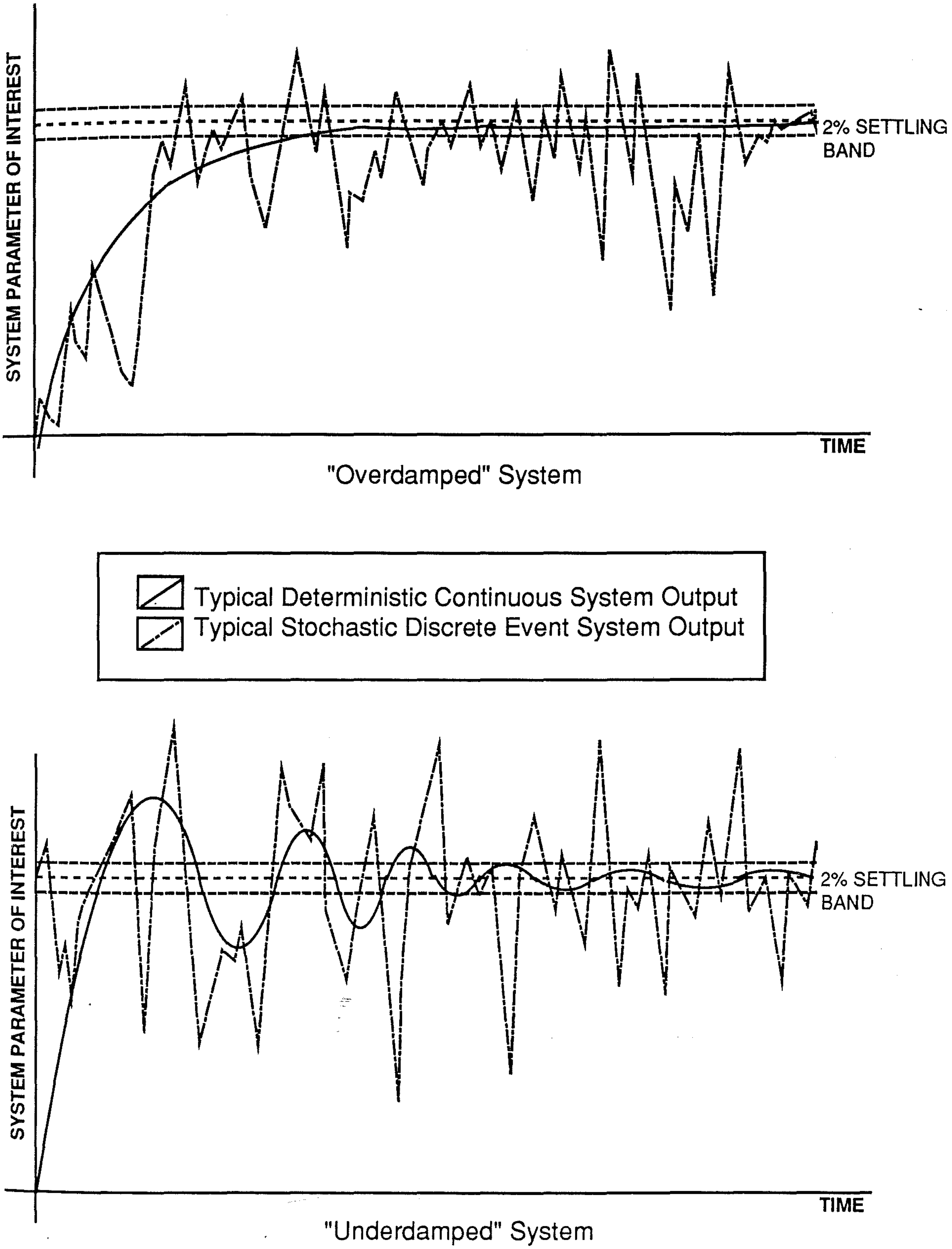
An entirely analogous definition is not possible for DEDS, or for stochastic systems in general. Figure 1.1 illustrates this point. The figure shows standard continuous-dynamic-system response functions, alongside output data points from "equivalent" discrete-event systems. Because the continuous functions are based upon systems of linear differential equations, settling time can be determined using basic linear algebra. Once the settling time has been reached, these deterministic functions always stay within the two percent (or five percent) settling band. DEDS output data points, on the other hand, exhibit random variability, even once the system has reached an apparent steady state. As the figure suggests, for an arbitrary steady-state distribution, observations in the steady-state range can be outside of any arbitrary settling-time corridor.

The settling time measure derived from continuous dynamic systems analysis is an appropriate concept for steady-state detection in discrete-event analysis, but the application of this measure is not as straightforward. In order to apply it to stochastic DEDS, a more robust definition of steady state is needed; one that is based on the same concept but can handle the probabilistic nature of discrete-event output data. Authors have used various approaches in attempting to develop such a definition. Before looking at these, some basic theoretical statistical concepts must be reviewed.

Clearly, the desired result of a DEDS study is a useful, meaningful estimator of the variable of interest. According to Fishman (1978),

> For $\hat{\theta}_n$ [a sample statistic based on n observations] to be a useful estimator [of the population statistic θ], in practice, we want $\hat{\theta}_n$ to converge to θ, in some probabilistic sense. When $\hat{\theta}_n$ satisfies
>
> $$\lim_{n \longrightarrow \infty} \Pr\{|\hat{\theta}_n - \theta| > \varepsilon\} = 0,$$
>
> where θ is the variable of interest, for arbitrary $\varepsilon > 0$, we say that it is a *consistent* estimator of θ, [and] the absence of consistency implies that additional information fails to improve one's knowledge of underlying phenomena.

Typical Continuous System Output Function vs. Typical Discrete Event System Output Function



"Overdamped" System

"Underdamped" System

Figure 1.1

In order for an estimator to be consistent, its mean square error and variance must vanish as the number of observations n increases. The parallel between the definition of deterministic steady state and that of a consistent estimator is clear. Measurement of the process x(t) over increasing time is replaced by measurement of the estimator θ_n over increasing observations. Since a typical estimator smooths the variations in the individual data points, it is reasonable to expect the variation in the estimator to decrease in some probabilistic sense as the number of observations increases.

For correlated observations, such as the output of a non-terminating simulation, consistent estimators may not be available unless the underlying stochastic process is *stationary* and *ergodic* (Papoulis, 1965):

> A stochastic process x(t) is stationary if its statistics are not affected by a shift in the time origin. This means that the two processes x(t) and x(t+τ) have the same statistics for any τ . . . .x(t) is stationary in the wide sense if its expected value is a constant and its autocorrelation depends only on $[t_1 - t_2]$.
>
> A stochastic process is ergodic if time averages equal ensemble averages (i.e., expected values).

In terms of simulation experiments, stationarity implies that for a fictitious replication of infinite length, the output statistics would be identical if data collection began at any arbitrary time during the replication. Ergodicity implies that the output statistics collected at any given simulation time are the same across all replications, for fictitious replications starting infinitely distant in the past.

It is easy to see the parallel between the concepts of steady-state behavior that we are attempting to define and the properties of stationarity and ergodicity. These properties exactly describe, on a theoretical level (for population statistics), the characteristics of steady-state data. Many of the definitions of steady state that can be found in the simulation literature rely on these properties, at least to some extent. Whether or not this is intentional is difficult to tell, because the properties represent a very "common sense" approach to the description of steady state. We consider two such definitions, one by Law and Kelton (1982) and a second by Gafarian *et al.* (1978).

Law and Kelton give the following interpretation of the transient and steady-state response in discrete-event simulation with respect to system delay time for an M/M/1 queue:

> Let $F_{i,\ell}(x) = P\{D_i \leq x | L(0) = \ell)$ [where D_i is the delay in the system at each time i]. We call $F_{i,\ell}(x)$ the *transient distribution of delay at time i* given $L(0) = \ell$. (The word "transient" means that there is a different distribution for each time i.) Now it can be shown that for any $x \geq 0$,
>
> $$F(x) = \lim_{i \to \infty} \{F_{i,l}(x)\} \quad \text{for } L(0) = \text{any } \ell$$
>
> exists, and we call F(x) the *steady-state distribution of delay*. . . . At the point in time when $F_{i,\ell}(x)$ is essentially no longer changing with i, we shall intuitively say that the process $\{D_i, i \geq 1\}$ is in 'steady state'. Thus, steady state does *not* mean that the *actual* delays in a single realization (or run) of the simulation become constant after some point in time, but that the *distribution* of the delays becomes invariant.

Constancy of the theoretical distribution of x_t obviously implies that all of the theoretical statistics of the distribution must be constant. This definition of steady state implies strict stationarity, i.e., that the marginal probability distribution function of x_t is independent of t (Fishman, 1978). In other words, Law and Kelton select the point at which the output data becomes strictly stationary as the onset of steady state.

Gafarian, *et al.* (1978) define the steady-state mean (μ_∞) the following way:

$$\mu_\infty = \lim_{t \to \infty} E[X_t | X_0] = E[X_\infty],$$

They state that the problem in steady-state detection is to find the minimum t, called t*, such that

$$1-\varepsilon \leq \frac{E[X_{t^*}]}{E[X_\infty]} \leq 1+\varepsilon,$$

where X_t is some discrete-parameter stochastic process being observed, and ε is some very small preassigned number. This means that steady state is the operating condition which occurs beginning at t^*, the time when the sample expectation of X_t is bounded by some arbitrary ε for all sample sequences beginning with X_0 and ending with X_t, $t \geq t^*$. The parallel with the definition of settling time for deterministic systems is clear, as well as with that of a consistent estimator. Since Gafarian *et al.* rely only on the first moment of the variable of interest and disregard other system statistics, however, their definition falls within the category of stationary in the "wide sense". Further, Gafarian *et al.* define the onset of steady state as the point at which sufficient data have been collected to debias the output statistic, achieving approximate stationarity in the wide sense, rather than the point at which wide stationarity is exhibited by the ensuing data.

Both of these definitions of steady state seem to be reasonable formalizations of the underlying ideal. The fact that these are different, and often result in different conclusions about the onset of steady state for the same data sample, reflects the elusiveness of the underlying concepts. In setting out to develop a methodology for defining and measuring steady state, this is one of the main issues that must be resolved.

1.4 MEASUREMENT OF STEADY STATE

In the preceding discussion, we identified the principal components of a theoretical definition of stochastic DEDS steady state and DEDS settling time. Operationally, the differences between alternative definitions of steady state imply that the simulationist must establish the sense of stationarity required for a specific application. This means that steady state must be defined not with respect to the output *variables* of interest, but with respect to the specific *statistics* of the underlying distributions of these variables that are thought to characterize steady-state operation. This determination largely will dictate the output data to be collected and, in so far as possible, should be made during exploratory simulation runs. Defining the statistics which characterize steady-state operation clearly requires a judgment, which must be made in consideration of both the anticipated behavior of the system and the purpose of the simulation study.

Differences in the theoretical definitions of settling time further imply that the simulationist must establish the alternative that will be used to remove the "start-up" bias from the output data. Operationally, the choices appear to be either *dilution* or *truncation* of the transient data, or perhaps some combination of the two. Generally, truncation is more efficient and, therefore, preferred, regardless of the purpose of the study.

To see this, consider a "well-behaved" sample, X_t, which has an invariant $F(X_t)$ for $t \geq t'$ and whose statistics approach steady state monotonically. For $\varepsilon << 1$, Gafarian *et al.*'s onset point, t^*, typically will be greater than Law and Kelton's onset point, t', for an arbitrary X_0. This is true because Gafarian *et al.*'s calculations are based on the cumulative mean, which is biased by the initial conditions, while Law and Kelton's calculations are not. Clearly, then, Law and Kelton's concept is preferable because it leads to a more efficient solution in which fewer observations are required for the same level of confidence in the results. The difficulty, of course, is in actually putting Law and Kelton's concept to use. The most desirable procedure for our purposes is one that uses the statistics of X_t without retaining the bias of the initial conditions, but gives more straightforward guidance than Law and Kelton's instructions to find "intuitively" the point at which the distribution becomes invariant.

The common components of the alternative theoretical definitions also have implications for steady-state measurements. All of the theoretical definitions of steady state quite properly involve the *invariance* of some output statistic *as time approaches infinity*. An operational definition of steady state, on the other hand, must recognize that simulations are inherently sampling experiments. Replications are of finite length. Output statistics will vary at some level of precision for different replication lengths, except in unusual cases. In the remainder of this section, we consider the transcendent operational issue of determining DEDS steady state and settling-time statistics (however defined) from actual simulation output.

Recognizing that DEDS simulations are sampling experiments, the key to an operational definition of steady state, involves the usual statistical notion of *confidence* in an estimate based on sample data. Consider, for example, Gafarian *et al.*'s definition of steady state. They say that steady state begins at $t=t^*$, when the mean of the data sample settles down to some error band. This assumes a finite sample with some initial condition, X_0, that fits some sample distribution, $F(X_t \mid X_0)$. Their reasoning seems to be that the smoothing effects of the cumulative mean calculations will eventually wash out the bias caused by X_0; therefore, for all $t \geq t^*$, $E[X_t|X_0]$, $E[X_{t^*}|X_0]$, and $E[X_{t^*}]$ are all approximately the same. But the only way to be absolutely sure of this is if the sample is infinitely large. Given a restricted sample size, the best we can do is to determine how much confidence there is that the mean has actually settled down.

Similarly, Law and Kelton say that steady state begins at $t = t'$, when the theoretical distribution $F[X_t|X_{t'}]$ is constant for all $t \geq t'$, with t' based on initial condition X_0. Again, to be 100% certain of its constancy, the sample must be infinitely large. Otherwise, we must settle for some level of confidence that t' has been found in the sample and that the distribution is constant for the observations after t'.

What becomes apparent in studying these two definitions is that there are two significant components necessary for a definition of steady state to be complete. One requirement is, obviously, initial detection of the settling of the transient response; the other is somewhat less obvious. This requirement is that the sample size, N, if not predetermined by external forces, must be large enough to affix a certain amount of confidence to the actual existence of steady state. This is equivalent to the determination of sample size for establishing system parameter confidence levels, and, in this sense, we treat the system settling time just as we would any other system parameter.

An example will clarify this concept. Consider a simulation in which steady-state system delay is the parameter of interest, and assume that onset of steady state has been estimated at point t^*. The sample is composed of all data points from the point t^* to the endpoint, N. Basic statistics says that a larger sample size results in higher confidence that the expectation from t^* to the endpoint closely estimates the theoretical limiting system delay value. At the same time, the larger sample size implies that the existence of steady state is being tested for a longer period of time. If transient data dominate the initial portion of the sample so as to make the sample *seem* to be in steady state, a larger sample size is more likely to detect the actual transient nature of the data. Therefore, increasing the sample size increases confidence in the existence of steady state, as well as increasing confidence in the calculated steady-state values. The inherent handicap of any time-constraining definition is that the systems involved are probabilistic; hence, any restriction placed on the run length limits the sample size of the data set. This changes the calculated values from population statistics into sample statistics, which immediately reduces the reliability of the solutions. This is why the level of confidence desired by the decisionmaker should be used in the determination of run length for simulations in which run length is not dictated by the simulation's purpose or some other condition.

Consider, on the other hand, a situation in which a decisionmaker is strictly interested in the steady state values of a given set of parameters based on the simulation of a two-month cycle of a manufacturing plant. It is clear that in this case the meaning of "steady state" may not exactly match its theoretical meaning and that run length must be based on the purpose of the simulation and the decisionmaker's needs. As long as the expectations are not transient throughout the entire set of data, the decisionmaker's concept of steady state has been found. Here, confidence levels are not as important as the decisionmaker's criteria for analysis. This case illustrates that the definition of steady state may be altered to fit the requirements of the situation.

Consequently, there are two ways to define the run length, N. The first is according to decisionmaker needs and simulation purpose, as in the above case. In this example, N is simply the number of data points generated by the simulation of a two month cycle of the plant. Here, statistics based on confidence levels are used only to confirm the existence of steady state, not to set run length. The second way to define run length, for the more general case in which the purpose does not strictly dictate N, is to base run length on the level of confidence the decisionmaker wants in (1) the existence of steady state, and (2) the estimated limiting values of his system parameters. Clearly, the confidence-level versus sample-size issue is of great importance in the development of an operational definition of DEDS steady state.

1.5 AN OPERATIONAL DEFINITION OF STEADY STATE

Based on the preceding discussions, we now offer an operational definition of stochastic DEDS steady state and settling time.

Let $\mathbf{X}_t$ be a vector of random processes, representing the DEDS simulation output variables of interest. Let $\theta(\mathbf{X}_t)$ be a vector of random processes, representing the output statistics which capture the sense of stationarity prescribed for the study. Let $\hat{\theta}(t_1,t_n)$ be an estimate of θ, based on a sample sequence $\mathbf{X}_t$, $t=t_1,\ldots,t_n,$ of n observations. Let $\pm\mathbf{Z}$ be an (approximate) vector confidence interval for θ at some prescribed confidence level. Finally, let $\mathbf{X}_0$ be the prescribed initial condition for the simulation.

For a given output sequence $\mathbf{X}_t$, $t = 0,\ldots,T$, the *sample settling time* $t^*(\mathbf{X}_0,T)$ is defined as the value of t which minimizes some prescribed norm of the confidence interval $\|\mathbf{Z}\|$ for the truncated sample $\mathbf{X}_t$, $t=t^*,\ldots,T$. It will be said that the sample process $\mathbf{X}_t$, $t=t^*,\ldots,T$ is in steady state if and only if $\|\mathbf{Z}\| \leq \varepsilon$, some prescribed minimum confidence norm.

To understand this definition on a more practical level, consider the processing of some device in a continuously-operating manufacturing plant. This plant has two constrained servers in parallel. For each server, once there are 30 devices in its queue, it "rejects" further devices until the queue shrinks. Let $\mathbf{X}_t$ represent the number of devices in the second queue at time t. Let $\theta\,(X_t)$ be the steady-state mean for the number in the second queue. $\hat{\theta}(t_1,T)$, the sample estimate of the mean, is then $E[X_t]$, where $t = t_1,...,T$ with t_1 to be determined. Assume that the initial state of this simulated process is empty and idle, that is, $X_0 = 0$, and let the total number of observations be, somewhat arbitrarily, 1875. For additional information on this model, see the SIMAN model files for the Filling Queue Model listed in Appendix A.

The sample settling time, as defined above, can be found by locating the point, t_1, at which the confidence interval is minimized; therefore, confidence intervals clearly must be determined. In order to compute statistically accurate confidence intervals, the data must be batched to remove the effects of autocorrelation resulting from the non-terminating nature of the model. For this example, confidence intervals have been assessed using an arbitrary batch size of 25. Figure 1.2 shows the behavior of the output function for this model. The graph displays the batched values based on the observations of X_t for $t = 1,...,\sim 9000$ minutes. Each point is the batch mean for a subsequent group of 25 observations. There are 75 points in all, corresponding to the total of 1875 observations.

In order to determine the sample settling time, t^*, confidence intervals are calculated using various values of t_1 until the minimum is found. Table 1.1 displays several confidence intervals computed based on the X_t sample (batched in groups of 25), with an increasing number of batches truncated before each computation. From both the table and the graph, it is clear that the confidence interval is minimized when five batches are truncated; therefore, the number of observations to be truncated is 125, and from looking at individual observations, it can be found that the associated time is approximately 450 minutes. This value is the sample settling time, t^*. Assuming that this situation requires a 95 percent confidence interval with a halfwidth size of less than two percent of the estimated mean, we can say that X_t is in steady state over the interval $t = t_1,...,T$ if and

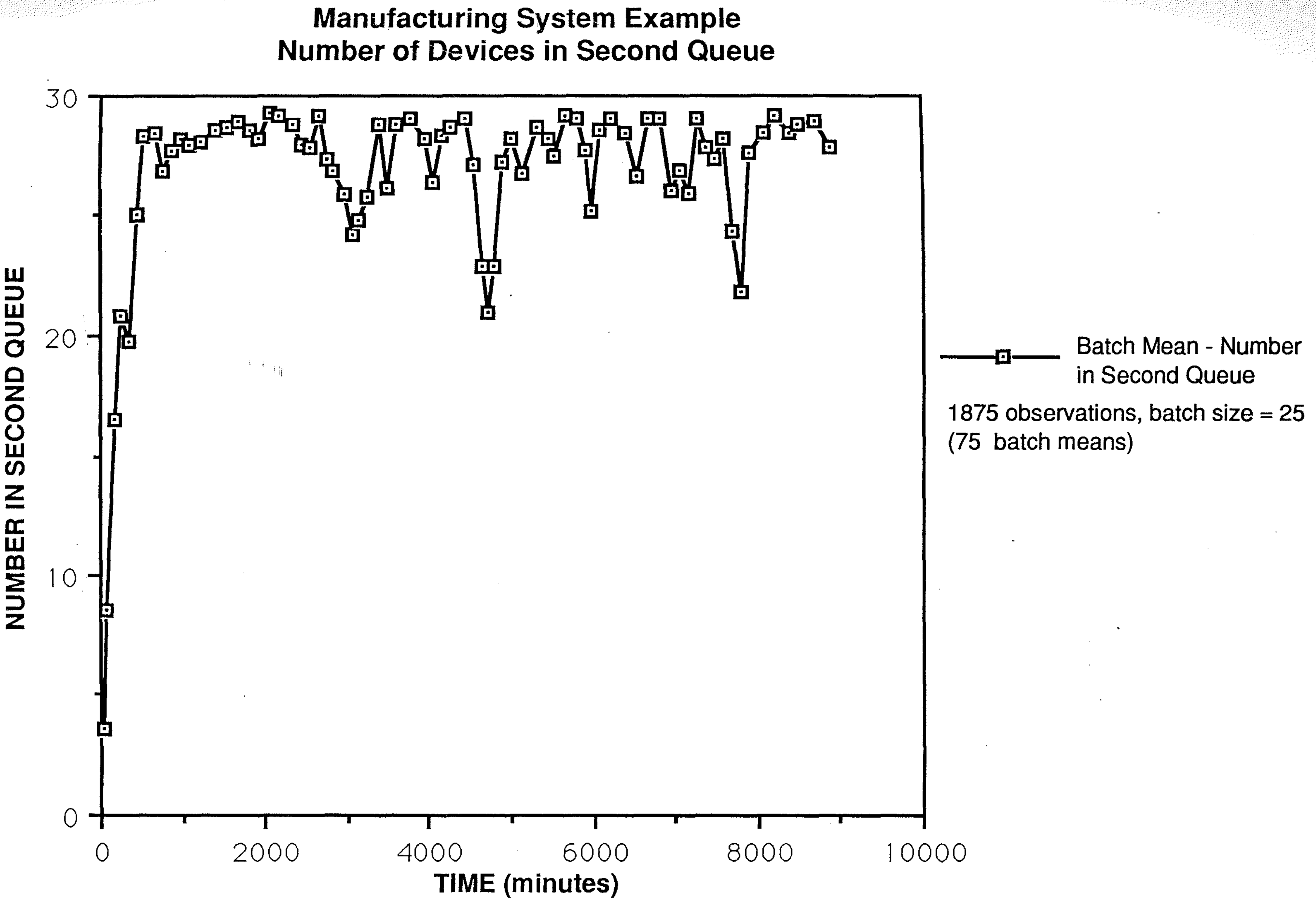

Manufacturing System Example
Number of Devices in Second Queue
30
20
10
0
NUMBER IN SECOND QUEUE
0
2000
4000
6000
8000
10000
TIME (minutes)
Batch Mean - Number in Second Queue
1875 observations, batch size = 25
(75 batch means)

Figure 1.2

Table 1.1
Effect of Initial Truncation on Confidence Interval for (Example) Manufacturing System

Batches truncated	95% Conf. Interval	Mean
0	0.9709	26.56
1	0.7595	26.88
2	0.5797	27.13
3	0.5070	27.27
4	0.4800	27.37
5	*0.4352	27.47
6	0.4358	27.51
7	0.4415	27.50
8	0.4474	27.48
9	0.4538	27.54
10	0.4607	27.49

only if $\|Z\| \leq 0.55$. In this case, for X_t from $t = \sim 450(t^*),\ldots,9000$ minutes, this requirement has been met, so X_t is said to be in steady state over this interval. If greater confidence were required, the sample size would have to be increased; in other words, T must be greater than 9000 minutes.

The operational definitions of steady state and sample settling time developed here provide a straightforward, quantifiable procedure for assigning a level of confidence to the existence of steady state in a given data sample. The power of this capability will be revealed by the development of a new steady-state detection heuristic and a methodology for evaluation of this and other detection heuristics in the remainder of this thesis.

1.6 THESIS OUTLINE

The next chapter will discuss the need for steady-state detection heuristics and will summarize the work that has been done by simulation researchers in developing procedures for evaluation of detection heuristics. Chapter Three will then provide a detailed methodology based on the definition that has been developed in Chapter One. Chapter Four will present the results of evaluation of some detection heuristics using the procedure developed in Chapter Three, and Chapter Five will discuss future research needs in this area.

CHAPTER TWO

PREVIOUS WORK IN STEADY-STATE DETECTION

2.1 INTRODUCTION

In this chapter, some of the previous work on steady-state detection heuristics will be reviewed. The review begins with the seminal work of Gafarian *et al.* (1978) and of Wilson and Pritsker (1978a,b) on the comparative analysis of alternative detection heuristics. In addition, we present the work of Schruben (1981) and the related work of Heidelberger and Welch (1983), who take a rather different approach from the earlier analysts. The similarities and differences among the various analyses will be discussed, as well as the strengths and weaknesses of each.

2.2 STEADY-STATE DETECTION HEURISTICS

A list of the most commonly used detection heuristics is provided here, in Table 2.1, as background to the information discussed in the body of this chapter. Each of these rules determines a point at which a sequence of simulation response data is commonly truncated for subsequent output analysis. The table includes both the heuristics evaluated by Gafarian *et al.* and the heuristics considered by Wilson and Pritsker. Also included in the table is a heuristic recently proposed by Ingalls (1987).

2.3 THE GAFARIAN ANALYSIS

In their seminal paper, Gafarian *et al.* developed a test procedure and set of criteria for evaluating alternative detection heuristics. Steady-state was defined with respect to the mean as

$$\mu_\infty = \lim_{t \to \infty} E[X_t | X_0] = E[X_\infty],$$

Table 2.1

COMMON TRUNCATION HEURISTICS

1. Conway/Data Interval Rule: Truncate the set of data until the first value is neither the maximum nor the minimum of the remaining set. Repeat for a few exploratory runs, then truncate future data sets based on the most conservative of the test runs (Conway, 1963).

2. Modified Conway Rule: Similar to Conway Rule, except that data is looked at in a backwards manner to find first observation that is not a maximum nor minimum of the previous values. Exploratory runs are used the same way as in Rule 1 (Gafarian *et al.*, 1978).

*3. Fishman/Crossings-of-the-Mean Rule: A running cumulative mean is kept as the data are generated, and the number of times the data crosses this mean is counted. When this number reaches a prespecified value, steady state is said to begin. Exploratory runs are not required for this rule (Fishman, 1973).

4. Gordon/Cumulative-Mean Rule: Run a pre-specified number of exploratory replications, each with a pre-specified number of observations and initial condition. Plot the grand cumulative mean over all observations and runs and choose a point at which the mean appears to stabilize. Truncate to this point (Gordon, 1969).

5. Gordon/Variance-Reduction Rule: Run a pre-specified number of exploratory replications to estimate the variance of the data values. Truncate up to the point at which the variance begins to fall off at a rate of 1/(sample size) (Gordon, 1969).

6. Schribner/Batch-Means Interval Rule: Divide an exploratory data set into batches and calculate each batch mean. Truncate to the point at which a specified number of the most recent batch means are within a specified interval of each other (Schribner, 1974).

7. Fishman/Autocorrelation Rule: Truncate the number of autocorrelated observations that is "equivalent" to one independent observation (Fishman, 1971).

*8. Emshoff & Sisson/Moving-Averages Rule: Divide the sample into "statistically large" subgroups (i.e., at least 30 if a z test is to be used). Truncate when there is no statistically significant difference between the average of the previous subgroup and the average of the current subgroup (Emshoff and Sisson, 1970).

*9. Ingalls/Cumulative-Statistics Rule: Truncate when both (a) the cumulative mean and cumulative standard deviation of sequential subgroups are within a specified range, and (b) the slope of the mean and slope of the standard deviation of sequential subgroups are less than a specified ratio (Ingalls, 1987).

*10. "Do Nothing" Rule: Retain all data. No truncation.

* No exploratory runs required

and the problem of steady-state detection was defined as that of finding the minimum t, denoted t^*, such that

$$1 - \varepsilon \leq \frac{E[X_{t^*}]}{E[X_\infty]} \leq 1 + \varepsilon.$$

These definitions were used to calculate t^* for entity waiting time (time-in-queue) in an M/M/1/∞ queuing system, using a variety of initial conditions and utilization factors (traffic intensities), the corresponding theoretical value of μ_∞, and error bands, ε, of 0.05 and 0.10. The queuing system was simulated to generate sets of test data from which the sample statistics for t^* were determined.

The first five rules in Table 2.1 were applied to the same output data. The truncation point determined by each heuristic was considered to be an estimate of t^*, denoted $\widehat{t^*}$, which is one possible value of a random variable $\widehat{T^*}$. The performance of each heuristic was judged by comparing the corresponding statistics of $\widehat{T^*}$ with those previously determined for t^*. Based on this comparison, heuristics were rated in terms of five criteria: accuracy, precision, generality, cost, and simplicity.

Accuracy measures how close the expectation of the estimator of t^* is to the actual t^*. Mathematically, accuracy is defined as:

$$\alpha = \frac{E[\widehat{T^*}]}{t^*}$$

where $\widehat{T^*}$ is the random variable estimator of t^*. High accuracy is attained if α is close to one. Precision is measured by the coefficient of variation of $\widehat{T^*}$:

$$p = \frac{\sqrt{Var[\widehat{T^*}]}}{E[\widehat{T^*}]}$$

Gafarian *et al.'s* idea of high precision is $p \approx 0$.

Generality means that a heuristic works well for a wide variety of systems. Cost is essentially a measure of the computer time required for:

(1) computational needs of the algorithm,
(2) collecting exploratory output data, if necessary, and
(3) making up any data lost from an overestimation of t^*.

Simplicity measures the accessibility of a heuristic to the average person. In the Gafarian study, all five heuristics were judged unsatisfactory and "not suitable for their intended use" because they each are judged to perform poorly in relation to one or more of these criteria.

While the importance of this pioneering analysis is undisputed, a number of fundamental problems invalidate the procedure employed and mitigate its sweeping rejection of the five heuristics tested. Most of these problems stem, directly or indirectly, from the use of the theoretical value of μ_∞ in the definition of steady state and settling time. Failure to use sample statistics and confidence intervals as the basis for evaluation limits the generality of the approach, leads to false notions of accuracy and precision, obscures tradeoffs between the confidence and cost of estimates, and generally precludes a meaningful comparison of the heuristics. We consider some of these problems in more detail in the following.

Most obviously, use of the theoretical mean in defining the experiment limits testing to processes for which the theoretical value can be calculated analytically, such as the M/M/1 queuing system actually employed. Such processes are comparatively rare and are not necessarily representative of the types of systems to which the detection heuristics are applied in practice. For example, the regenerative nature and high variability of the M/M/1 queue makes this simple process one of the most difficult in which to detect steady state.

Gafarian *et al.* defend the use of this test system based on the generality criterion, stating that heuristics which break down for M/M/1 do not merit further investigation.

> These rules will no longer be tested in other situations, since, even if they produced good results in some other cases, our criterion for generality would not have been met.

In fact, it is the testing condition that fails to generalize. Heuristics that are rejected in this instance should be tested on a wide variety of more typical processes, for which the theoretical mean is estimated from the sample data actually used by the heuristic.

More importantly, using the theoretical mean as the basis for evaluating the accuracy and cost of a heuristic obscures the individuality of the experimental data on which the heuristic actually operates. Because the analysis is based on a sampling experiment, it is unlikely that the theoretical value is in fact the best estimate of the mean or settling time that can be had from the data, i.e., the sample steady-state mean and sample settling time defined in Chapter One. Furthermore, the sample data support only limited confidence in the accuracy of the estimates, which must be determined as a part of the experiment. Using the theoretical mean (inadvertantly) implies 100% confidence in this value, which is not supported by the data. To salvage the Gafarian methodology would require prohibitively long test sequences, in order that the theoretical mean and implied confidence in this value begin to approximate the true sample statistics. This point is perhaps as subtle as it is fundamental.

As a consequence, Gafarian *et al.* improperly reject three of the five heuristics tested primarily because these overestimate t^*. We note, first, that Gafarian *et al.'s* judgment of conservatism is based on an arbitrary choice of ε and the misuse of the theoretical mean in calculating t^*. Moreover, conservatism in itself is not an appropriate rejection criterion. Truncating a longer sequence of data provides greater certainty that steady state has been achieved in the data remaining. A more appropriate way to assess acceptability is to weigh the extra confidence in the achievement of steady state gained by a "conservative" t^* against the cost of the additional data required to achieve comparable confidence levels.

For a fixed sequence of data, as used in the analysis, our definition of the sample settling time provides a means for quantifying this assessment and measuring exactly how good or how bad a heuristic estimate is. A conservative truncation point is one which is greater than the sample settling time; a premature truncation point is one which is less than the sample settling time. Since the number of data points is fixed, the cost of a heuristic is some appropriate combination of (1) the loss of confidence in the resulting estimate of the mean, i.e., $|Z - \hat{Z}|$, where Z is the confidence interval for the sample mean and $\hat{Z}$ is the confidence interval of the heuristic estimate of the sample mean, both defined for the sample confidence level; and (2) the increased or decreased computation time of the heuristic, compared with the time required to compute the sample settling time. The accuracy of the heuristic with respect to estimating the sample mean is the difference between the sample mean and the heuristic estimate of the sample mean, i.e., the bias $|E[X_{t_s}] - E[X_{\hat{t}^*}]|$, where the sequence X_{t_s} runs $\{t_s,...,T\}$, for the sample mean, and the sequence $X_{\hat{t}^*}$ runs $\{t^*,...,T\}$, for the heuristic estimate. The accuracy of the heuristic with respect to estimating the sample settling time is the number of data points between the sample settling time and the heuristic truncation point, i.e., $|t_s - \hat{t}^*|$.

Using these measures, a consistent comparison of alternative heuristics can be made, with respect to both accuracy and cost. Indeed, these measures can be used to measure the accuracy and cost of Gafarian *et al.'s* t^* as an estimator of t_s and to measure the consequences of the arbitrary choice of ε and misuse of the theoretical mean in the analysis.

While use of the theoretical mean in the Gafarian analysis confuses the evaluation of accuracy and cost, the analysis also presents problems with respect to the criterion of precision. One of the reasons some of the rules are rejected is that the precision values are unacceptable; that is, they are not close enough to zero. In Gafarian *et al.'s* interpretation, precision is a measure of the variation in the set of t^* estimates produced by a particular heuristic. This sample of t^* estimates is the result of execution of the heuristic for several output sets of a given simulation; i.e., several runs of the M/M/1 queue model simulation, each with a different random number seed. Gafarian *et al.'s* requirement is that there be almost no variation among these t^* estimates. The problem is that, because of the natural variation of the underlying distributions of the output data, it is very probable that the actual t^* values have a high degree of variation themselves; therefore, it is illogical to require the t^*

estimates to exhibit low variation. It simply does not make sense for Gafarian to use the precision criterion in evaluating the effectiveness of a truncation heuristic.

Based on the problems brought out in the above discussion, it is safe to say that Gafarian *et al.'s* sweeping rejection of the first five truncation heuristics is premature. Additional research with extra emphasis on statistical comparisons is required before any such conclusions can be made.

We note, finally, that the selection of a "best" heuristic using our method is a multiobjective problem, just as it should be in the original Gafarian analysis. The difficulty here is in assigning tradeoff weights to the various measures of accuracy and to the component measures of cost. It is likely that no single heuristic will dominate with respect to all of these criteria and that the selection of a heuristic will depend on the purpose and constraints of the simulation study. The improved evaluation method will inform this decision. This result clearly is preferable to a sweeping rejection of all heuristics.

2.4 THE WILSON AND PRITSKER ANALYSIS

In two papers published soon after the Gafarian analysis, Wilson and Pritsker present an alternative procedure for evaluating steady-state detection heuristics. The first paper (1978a) surveys research on the so-called "startup problem". Three approaches to the problem were found in the literature: time-series analysis, queuing theory, and detection heuristics. Wilson and Pritsker conclude that

> Although the results derived from time-series analysis and queuing theory are rigorous and precise, they have rather limited applicability. On the other hand, many of the heuristic methods have broader applicability but are ambiguously formulated and have uncertain statistical properties.

Several prior analyses of detection heuristics are briefly reviewed, including the Gafarian *et al.* study. Wilson and Pritsker correctly note that Gafarian *et al.*

> did not examine the full effects of truncation on the estimator of μ_X [the theoretical mean]. The best policy for estimating μ_X may not necessarily also be the best policy for estimating t^*.

They further conclude that an evaluation procedure is needed that

> focuses directly on the behavior of the truncated sample mean, ...consider[s] the random variation in the truncation point, [and] characterize[s] both the random and systematic components in the estimation error.

Such a procedure is developed and applied in the second paper (1978b).

Statistics were generated for number-in-system for 50 observations of a finite capacity single-server queue (M/M/1/15) and a finite capacity machine-repair queue (M/M/3/14/14) from the theoretical probability transition functions for these stochastic processes. These statistics were based on three different initial conditions:

(1) "empty and idle";
(2) X_0 as close as possible to the theoretical steady-state mode; and
(3) X_0 as close as possible to the theoretical steady-state mean.

For each of the three, the bias

$$B[\overline{X}_{50,d}|X_0{=}i] = E[\overline{X}_{50,d}|X_0{=}i] - \mu_x ,$$

variance

$$V(\overline{X}_{50,d}|X_0{=}i) = E([\overline{X}_{n,d} - E(\overline{X}_{n,d})]^2|X_0{=}i),$$

and mean square error

$$\begin{aligned} MSE(\overline{X}_{n,d}|X_0{=}i) &= E[(\overline{X}_{n,d} - \mu_x)^2|X_0{=}i] \\ &= V(\overline{X}_{n,d}|X_0{=}i) + B^2(\overline{X}_{n,d}|X_0{=}i), \end{aligned}$$

were tabulated for all 50 possible truncation points, d = 0,...,49, where $\overline{X}_{50,d}$ is the mean of the truncated sample and μ_x is the theoretical mean. These bias, variance, and mean square error values represent the *a priori* expected statistics of the sample runs.

Four detection heuristics from Table 2.1 were considered: the Do-Nothing rule (dilution), the Conway rule, the Crossings-of-the-Mean rule, and the Batch-Means rule. For each combination of heuristic and initial condition (termed a "startup policy"), empirical probability distributions for the recommended truncation point were developed from independent simulation replications. The average bias, variance, and mean square error for each startup policy was computed from the tabulated data by using an estimated probability distribution of the truncation point, d, over independent simulation runs. These results in turn were used to construct normalized confidence intervals for the sample steady-state mean for each policy. Finally, policies were compared based on the average sample confidence interval coverage of the theoretical mean. Based on these comparisons, it was concluded that

> the judicious selection of an initial condition is more effective than truncation in improving the performance of the sample mean as an estimator of the steady-state mean.

The methodology developed by Wilson and Pritsker is free of the conceptual errors that plague the Gafarian analysis. In order to understand Wilson and Pritsker's results better, some of the underlying effects of their methodology must first be explained. The basis of their entire methodology is the use of theoretical probability transition functions for the determination of bias, variance, and mean square error. These data represent the *expected* values of bias, variance, and mean square error, associated with the given initial conditions, as the number of replications approaches infinity. Because the sampling process is theoretically-based, its mean value progresses smoothly from the initial condition to the theoretical mean value of the underlying process as the sample size increases. Assuming that the initial conditions are not "unusual" for steady-state operation (which is an acceptable assumption for their study--this will be explained later), this smooth progression means that the expected bias will consistently decrease and the expected variance will consistently increase throughout truncation. Bias decreases because truncation eliminates observations that on average are skewed toward the initial condition. Variance increases because truncation reduces the number of observations. Given that the initial conditions are within normal data range, the truncated observations on average are not different enough to significantly affect the variance. Hence, Wilson and Pritsker's analysis indicates, correctly, that in certain situations, there is a compromise between bias and variance reduction as initial observations are truncated.

Their conclusion that truncation is an inefficient method of obtaining good statistics follows from this compromise. Given that their recommended approach is to choose an appropriate initial condition, we may assume that a "pilot run" is required before simulation data is actually collected. Because of the compromise between bias and variance, if a heuristic is used on the pilot run to determine the truncation points for all subsequent runs, it is likely for some processes that "good" data will be thrown away. This is especially true if the initial condition used for production runs is the mean or mode of the pilot run because this value is even more likely to represent a "good" initial condition; hence, the increase in variance due to truncation will be significant.

As indicated above, however, there are some restrictions associated with Wilson and Pritsker's results. First of all, their method relies on multiple replications of short runs, whereas the preferred way of analyzing non-terminating simulation output is to use data from a single long run. In addition, observed data are random, whereas the theoretical statistics are deterministic. The observed data do not necessarily progress monotonically from the initial condition to the population mean value and are not bounded by either the initial condition or the population mean (in a limited sample). Thus, it is always possible that statistical estimates from one long simulation run may actually be improved by truncating observations. Finally, Wilson and Pritsker's methodology requires "reasonable" initial condition values to be known at the start of the production runs, and this requirement may not always be achievable.

The link between Wilson and Pritsker's results and our approach lies in the relationship between sample settling time, as defined in Chapter One, and the behavior of the bias, variance, and mean square error. The data and measures developed in the Wilson and Pritsker analysis demonstrate that the sample settling time is an optimal truncation point in the absence of foreknowledge of an acceptable initial condition value. For the tabulated data, truncation at the sample settling time (zero, in the case of their examples) does in fact result in a minimum or near-minimum square error in all data sets. The reason for this is clear from the MSE equation, where the error criterion is shown to be the sum of the sample variance and the square of the bias.

If a "judicious selection" of the initial condition can be made, the bias term will be small relative to the variance term, the sample settling time will always be near zero, and truncation at the sample settling time (i.e., no truncation) will lead to a near minimum estimation error for a sufficiently large sample. On the other hand, if there is no basis for making a "judicious selection", then there is no prior knowledge of the bias, and the best strategy is to minimize the variance by truncating at the sample settling time. In effect, truncation at the sample settling time can be viewed as one means for making a "judicious selection" experimentally, since the observation corresponding to truncation point becomes the initial condition for the truncated sequence. The cost of the information concerning a good initial condition is the cost of the data that is rejected.

2.5 THE SCHRUBEN/HEIDELBERGER AND WELCH ANALYSIS

Our final analysis concerns a methodology for controlling simulation run length in the presence of an initial transient, described in a 1983 paper by Philip Heidelberger and Peter D. Welch. The transient detection aspect of their methodology is based on an approach described in a 1981 paper by Lee W. Schruben. Because our interests concern both transient detection and the "cost", or run length requirements, associated with initialization bias, we will look at both of these papers. First Schruben's approach to transient detection will be described, then it will be placed in the context of Heidelberger and Welch's overall methodology.

Schruben's approach is somewhat different from the others reviewed here. First of all, the purpose of his methodology is to determine only whether or not initialization bias exists in a given set of simulation output data. He does not attempt to identify where the bias ends within the data set. Heidelberger and Welch extend Schruben's concept to include identifying the location at which the bias disappears. Schruben's method attempts to standardize "the stochastic process being simulated so that it represents 'noise' in which a 'signal', due to initialization bias, may be detected." Schruben's standardization process is conceptually similar to the standardization process used in applications of the classical central limit theorem; that is, the procedure attempts to find a limiting distribution for the test statistic. The major difference here, though, is that instead of using a limiting normal random variable to standardize, as in classical central limit theorem applications, Schruben's method uses a limiting stochastic process, the standard Brownian bridge. His procedure is essentially a process version of the central limit theorem.

In order to perform this standardization, the output series is divided into the "noise" function, X, which is stationary, or unbiased, and the "signal", μ, which is affected by initialization bias. In other words, if Y_i is the actual value of a process at time t_i, then $Y_i = \mu_i + X_i$, where X_i is purely a function of the state of the system between time t_{i-1} and time t_i, and μ_i is the amount added (or subtracted) by initializing and running the simulation.

Schruben's determination of whether or not initialization bias exists is based on the sequence of partial sums,

$$S_n(k) = \overline{Y}_n - \overline{Y}_k;\ k = 1,2,\ldots,n,$$

where $\overline{Y}_n$ is the average of the entire output series, and $\overline{Y}_k$ is the average of the first k observations.

By manipulating the relationships between these entities and relying on some basic statistical properties, Schruben determines the standardized noise function and the standardized signal function. Combining these two functions creates a "standardized test sequence",

$$T_n(t) = \frac{[nt]S_n([nt])}{\sqrt{n}\sigma};\ t \in [0,1],$$

where n is total number of observations and t is time scaled to the unit interval (t = 1/n, 2/n,...,1). Schruben asserts that initialization bias can be detected by the existence of a prominent peak in this function that occurs at a relatively small value of t. In other words, assuming that the simulation user knows the sign of the bias, $T_n(t)$ is not expected to have a large positive maximum value if there is no negative initialization bias. If positive bias is expected, the output series is multiplied by (-1), and a similar expectation is true.

Schruben develops a test procedure based on the fact that if no initialization bias exists (i.e., μ_i is constant throughout the run), then $T_n(t)$ can be modeled as the standard Brownian bridge process, $\{\beta_t; t \in [0,1]\}$. By working with the joint density of t^*, which is the location of the maximum of β_t, and s^*, which is β_t^*, Schruben sets up several variable definitions from which his test procedure is developed. The procedure uses a hypothesis test in which the null hypothesis is that the output process has a constant mean. The value of $\hat{\alpha}$ for the hypothesis test is determined by manipulating the observed statistics and using an F distribution with three and ν degrees of freedom. The values of $\hat{\sigma}^2$ and ν, which are necessary for the test procedure, can be estimated using autoregression techniques. Schruben's methodology here is based on previous work by Fishman (1973). Once $\hat{\alpha}$ is computed, it is used to assess whether or not a test statistic more unusual than that observed will occur if there is no initialization bias present. If $\hat{\alpha}$ is large, the output probably does not contain a significant negative initialization bias. The "no negative bias" hypothesis is rejected if $\hat{\alpha}$ is less than the specified probability, α, of rejecting a true hypothesis.

Using this procedure, Schruben tests sets of simulation output data from five different models, with and without initialization bias. The behavior of the power functions of $\hat{\alpha}$ is monitored as α increases from zero to one. The models he uses have known steady-state distributions; therefore, he knows where to anticipate bias based on the initial conditions he uses in each run. According to Schruben's test results, his detection procedure is highly effective for a wide variety of simulation models.

Schruben calls attention to the test of an M/M/1 queue system as one notable "exception" to the success of his procedure. In the run of the M/M/1 queue with "empty and idle" initial conditions, Schruben seems to expect a strong indication of negative initialization bias from his procedure because a waiting time of zero is clearly less than the steady-state mean waiting time. The procedure gives a fairly weak indication of bias, and Schruben interprets this as bad performance of the procedure. However, when one recalls that zero is not an unusual value for the cyclic, highly variable M/M/1 queue, it becomes apparent that Schruben's assessment of the procedure's performance in this case may have been unnecessarily harsh. In other words, starting an M/M/1 queue as "empty and idle" does not actually introduce a strong negative initial bias, as Schruben seems to think.

In general, Schruben's method seems to work well in the act of bias detection; however, since it does not attempt to locate the ending point of the bias nor consider required run lengths, it is difficult to assess the usefulness of Schruben's method in the context of actual simulation output analysis. Soon after Schruben's results were published, Heidelberger and Welch (1983) published their work on run length control in the presence of an initial transient, which places Schruben's detection methodology into a more practically useful procedure. The purpose of their procedure is to determine the run length required to attain a pre-specified confidence interval and to incorporate the transient test so as to optimize the output statistics (e.g., maximize confidence while minimizing run length).

Heidelberger and Welch state that they wished to design a methodology to be useful to the "wide population of experimenters who have little knowledge or interest in simulation output analysis". They feel this can be done by incorporating a fairly complex set of procedures in a high-level simulation package that could be transparent to the user. A simple user interface would require the user to specify a few parameters.

> Ideally, these parameters should enable the users to define the accuracy they require and the maximum amount of computing time they are willing to invest, but *not* involve them any further in the technical details of the procedure.

As stated previously, Heidelberger and Welch use a method of initial transient detection based on Schruben's Brownian bridge model test. Their procedure uses Schruben's test a number of times to identify the transient portion of the data, then generates confidence intervals based on the remaining, stationary portion of the data. The confidence intervals requirement set by the simulationist is used to determine dynamically the optimal run length.

The procedure itself is as follows:

0. Set four parameter values:
   - j_{max}, maximum number of observations (run length)
   - j_1, initial checkpoint
   - I, multiplicative checkpoint increment parameter
   - ε, relative half-width requirement.

1. Perform stationary portion testing on data up to checkpoint j_k (initially, $k = 1$) to find n_0, if it exists, where $\{X(n), n = n_0+1,...,j_k\}$ is a sample from a covariance stationary process:

   a) Test $\{X(n), n = 1,...,j_k\}$ to determine if there is initialization bias. If no, then $n_0=0$. If yes, then (b).

   b) Remove initial 10% of the data and repeat (a) (now $n = [j_k/10]+1,...,j_k$). If no initialization bias, then $n_0 = j_k/10$. If yes, repeat (b).

   If no such n_0 can be found, then j_k becomes $j_{k+1} = \min\{I * j_k, j_{max}\}$. Repeat Step 1.
   If j_{max} is reached and no n_0 is found, then no confidence interval for μ can be formed.

2. Once n_0 is located, generate a confidence interval from $\{X(n), n = n_0+1,...,j_k\}$.
   Find estimated relative halfwidth (ERHW) of the C.I.:

$$\text{ERHW} = \frac{\text{C.I. halfwidth}}{2*\overline{X}}$$

$$\text{where } \overline{X} = \sum_{n=n_0+1}^{j_k} X(n)$$

   If ERHW $\leq \varepsilon$, then simulation may stop.
   If ERHW $> \varepsilon$, then j_k becomes $j_{k+1} = \min\{I*j_k, j_{max}\}$.
   If j_{max} is reached, the confidence interval generated for μ may or may not satisfy the accuracy requirement ε.

Heidelberger and Welch used their procedure to determine optimal truncation and run length for three different stochastic processes and four different initial transient functions. The initial transients they used were deterministic functions of various sizes and strengths which were simply added to the (known) stationary output series. With these models, they looked at the procedure's effects on confidence interval coverage, point estimate bias, mean run length, and mean amount of data truncated. In addition, they studied the effects of size and shape of the initial transient and the correlation structure of the output data.

Heidelberger and Welch performed tests initially to compare the relative effectiveness of four initial transient detection tests based on four alternative statistics that can be used for the Brownian bridge test. From these tests, they determined that the behavior of the run length control procedures is very similar, regardless of which of the four transient tests is used. For a given accuracy criterion, all four tests produce point estimates with very low relative bias, generally 3% or less; confidence intervals have approximately the same coverage of 0.90, which is, appropriately, the prespecified level of the transient tests; mean run length and mean amount of data truncated are also approximately the same over all transient detection tests. Based on these results, the Cramer-von Mises test was chosen because it is the simplest to use. In addition to the transient tests, further tests were performed to study the impacts of changing various parameters, including test levels and location and size of the initial checkpoint (j_1).

Based on the results of these various tests, Heidelberger and Welch note that, in general, the introduction of a transient test improves performance of their run length control procedure. With the use of a transient test,

> there is only a slight loss in coverage in the no transient case, and no increase in run length. In the cases where there is a transient, the coverage remains adequate with the introduction of the transient test and the run lengths are much shorter. The coverages for the case when no transient test is applied are way off.

They also conclude that their procedure performed very well when faced with a strong transient and when the initial checkpoint (j_1) is beyond the end of the transient phase. However, it was not as effective in detecting a weak transient or in detecting a transient when the initial checkpoint was within the transient period.

Heidelberger and Welch's technique for combining transient detection and confidence interval testing to determine run length is very effective and has some very specific advantages over procedures studied and developed in previous papers. For example, it is based on the use of one long simulation run, as most non-terminating simulation output sets are created. In addition, it makes things fairly easy for the simulationist by requiring nothing more complex than accuracy requirements and run length constraints to be input, and it does the rest automatically. It also allows the amount of data needed for final calculations to be as little as the accuracy requirements will allow.

The only real problem with Heidelberger and Welch's procedure is that it may be somewhat computationally intensive. Schruben's transient detection method does not have extreme computing requirements in itself for a single run: Schruben states that "order n storage locations" and "insignificant computation" is required for the first step of his procedure; in addition, two of the estimators may be taken from a table in Fishman (1973) or obtained using a modification of a subroutine by Fishman; and a probability value may be taken from F-distribution tables. Although each step in itself does not seem overwhelming, when used within Heidelberger and Welch's cyclical procedure, the whole process will have to be performed many times for every piece of every output series (in the worst case). At best, the computational requirements may call for the use of a fairly powerful computer; at worst, they may cause a computer to take an unreasonably long time to complete an analysis of the simulation output.

Finally, although this may not necessarily be considered a disadvantage, it should be noted that Heidelberger and Welch's procedure, because of its relative complexity and its unique transient detection method, is not easily understood by the average simulationist. The procedure *must*, without a doubt, be almost fully automated. The potential problem therein is that because there is less interaction and understanding by the user, there is also less chance that he/she will be aware of any existing problems with the analysis done by the computer. However, since this is clearly a problem with automation in general, it should be considered beyond the scope of this research.

2.6 CONCLUSIONS

In this chapter, we reviewed the prior literature on steady state/initialization bias detection methods and heuristics, as well as evaluation methodologies for these heuristics. We focused on the seminal detection heuristics evaluation work of Gafarian *et al.* (1978), the "startup policy" evaluation work of Wilson and Pritsker (1978), and the initialization bias detection and run length control methodology developed respectively by Schruben (1981) and Heidelberger and Welch (1983).

We showed that the Gafarian paper, while very enlightening as the first major work in the area, has serious gaps in the concepts and evaluation procedures used. Gafarian *et al.'s* most serious error was the use of theoretical rather than sample statistics as the basis for evaluations and comparisons. Wilson and Pritsker's work was shown to be both enlightening and robust as a description of the theoretical workings of initialization bias effects and some previously developed heuristics. However, Wilson and Pritsker's developments are not particularly useful for the execution and analysis of a "real-world" simulation.

Finally, we outlined the initialization bias detection method of Schruben and the run length control procedure developed by Heidelberger and Welch that incorporates Schruben's detection method. In this section, it was shown that, although the procedure may be somewhat computationally intensive, it could be very useful for automated simulation output analysis. It requires only empirically obtainable statistics, so it does not require foreknowledge of system statistics. In addition, it is based on theoretically sound statistical relationships among confidence interval width and coverage, bias, amount of initial data truncated, and run length.

In the next chapter, we will use some of the measures and definitions outlined here to develop a new heuristic for determining the optimal truncation point for non-terminating simulations. In later chapters, we will use these measures and definitions to compare the efficiency and effectiveness of this heuristic against some of the steady-state detection heuristics that were described in this chapter.

CHAPTER THREE
A NEW METHODOLOGY

3.1 GENERAL METHODOLOGY

As Chapter Two indicated, there are various problems associated with current methods for determining the onset of steady state and for evaluating detection heuristics. This chapter will describe a new evaluation procedure that is based on the steady-state concepts presented in Chapter One and that bears some resemblance to Heidelberger and Welch's concepts outlined in Chapter Two. In addition, a new steady-state detection heuristic that derives from this evaluation procedure will be introduced.

Recall that a new, operational definition of the steady-state detection problem was presented in Chapter One. This definition assumes a given initial condition and initial maximum confidence interval and considers detection of steady state as the problem of finding the "sample settling time", which is the point at which truncation maximizes the confidence level of the sample mean estimate. This definition of steady state is operationally feasible, as opposed to previously-used definitions, because it relies on confidence intervals and statistics based on finite samples of data. Note that confidence levels and confidence intervals have an inverse relationship. An increase in confidence level with constant halfwidth is equivalent to a decrease in confidence interval (CI) halfwidth with confidence level held constant.

The heuristics evaluation procedure consists of three steps:

1) Set a "base" CI halfwidth by obtaining a sample of data that can be collected in what the simulationist considers "a reasonable time" (a minimum of 1000 observations is generally required). The CI calculation should include all data (i.e., initial transient points should not be removed).

2) Truncate points from the beginning of the data set and study the behavior of the confidence level/interval as the number of truncated points increases. Truncation should end when the confidence level begins to decrease (or CI halfwidth begins to increase).

3) Compare the confidence levels achieved by other truncation heuristics against the statistics achieved by this confidence maximization-based truncation point.

For a well-behaved process with transient bias, the CI halfwidth will decrease, or confidence level will increase, as points are truncated until truncation enters the steady-state region of the data. At the onset of steady state, the confidence level will peak and then begin to decrease. The time of the confidence level peak approximates the sample settling time and is, therefore, the optimal truncation point.

The rationale behind this procedure is that two quantities have the greatest amount of influence on the confidence of a sample statistic: sample variance and number of observations, or sample size. Recall the definition of the 100(1-α)% confidence interval for the mean of a Normal population:

$$\left[\overline{X} - \frac{sZ_{\alpha/2}}{\sqrt{n}},\ \overline{X} + \frac{sZ_{\alpha/2}}{\sqrt{n}}\right],$$

where α is confidence level, $\overline{X}$ is sample mean, s is sample standard deviation, n is sample size, and Z is the Z-statistic value at a confidence level of (1-α). Using the Central Limit Theorem, it can also be said that this definition holds true for samples that are not necessarily Normal but are "very large". We will assume that our samples contain at least 1000 observations, so this definition may be safely applied. The above definition shows that the size of the CI is directly proportional to the sample standard deviation and inversely proportional to the square root of sample size. Therefore, a decrease in the variability within a set of data will cause a decrease in the size of the CI, while a decrease in sample size will have the opposite effect--the size of the CI will be increased.

A theoretical explanation for the peaking behavior of the confidence level at the onset of steady state can be found in Wilson and Pritsker's work (1983a,b), introduced in Chapter Two. Recall that Wilson and Pritsker studied the effects of initial truncation on the theoretical bias and variance of sets of output data from different models with a variety of initial conditions. For the data sets they tested, truncation of initial observations consistently caused the bias to decrease and the variance to increase. This effect occurred because, in every case, the initial conditions they used were not outside of the "normal" steady-state range of values for their models. Bias reduction occurred because, regardless of the "normality" of the initial condition, there will always be a slight skew in the initial output toward the initial condition values (because the initial conditions must be natural numbers, while the theoretical steady-state mean is non-integer). On the other hand, variance consistently increased because the sample size reduction caused by truncation had a much greater impact on the statistic than did the slight reduction in bias.

Although Wilson and Pritsker's testing did not specifically cover the case in which initial condition values are well outside the range of "normal" steady-state values, ideas from their analysis can be extrapolated to cover this case. Clearly, the initial data generated by a model run with very unusual initial conditions will be much more strongly skewed than the previous case; hence, bias caused by the initial conditions will be large. As initial observations are truncated, the statistics (for a well-behaved, converging function) will move closer to steady-state values, which will cause the bias to decrease. Assuming that the sample size is statistically large (which is not true of Wilson and Pritsker's test cases), the variance will not immediately be strongly affected by reduction in sample size. The variance will, however, decrease in response to the bias reduction caused by removal of outlying data points.

Once truncation reaches a point at which the remaining data values are close to steady-state values, the bias reduction will slow and its effect on the variance calculation will fade away. In addition, the effect of removing observations will eventually become stronger and will cause the variance to increase.

The truncation problem, then, is to determine how to assess the balance between bias and variance so as to find the optimal truncation point. Intuitively, it seems that the optimal point should be the point at which variance begins to increase. However, initial testing (as will be seen in Chapter Four) indicates that judging by the variance tends to produce overly conservative truncation points. Apparently, the effects of bias reduction are more powerful than sample size reduction in computing the variance; therefore, sample size must be weighted more heavily in determining optimal truncation point. This can be achieved in a rather straightforward manner by studying the behavior of the CI halfwidth instead of the variance. Since the CI calculation divides the statistic by the sample size one more time, the sample size gets the extra weight it needs. The theory behind this effect should be studied at some point to determine the cause; however, this is somewhat outside of the scope of this research. It will do for the time being simply to state that, empirically, the CI halfwidth seems to be a more effective statistic to use in determining the optimal truncation point.

Clearly, the CI halfwidth will exhibit behavior similar to the variance as described earlier; that is, it will decrease as a result of the bias reduction from truncation of outlying data values; then, it will increase once bias reduction slows and decreasing sample size becomes more powerful. The point at which the CI halfwidth reaches a minimum is the point of maximum confidence for the output statistics. This is the optimal point at which to stop truncation. This maximization of the confidence level (or, equally, minimization of CI halfwidth) approach is the theoretical basis of the heuristics' evaluation procedure; thus, it will be called the Confidence Maximization Procedure.

3.2 THE PROCEDURE

The Confidence Maximization Procedure (CMP) will now be described in detail. Most of the required calculations can be performed using the SIMAN (Simulation Language) Output Processor.

The first step is to run the simulation for a "conveniently long" time (according to the simulationist's best estimate; again, a minimum of 1000 observations is generally required) and to collect data for the system variable of interest (e.g., time in system, number in system, number in queue, etc.). Next, the "initial confidence" of the sample is determined. This is done by calculating the (1-α)% CI halfwidth:

$$\frac{sZ_{\alpha/2}}{\sqrt{n}} ,$$

where α = (1.0 - confidence level), based on all available data (no truncation).

At this point, truncation begins. Truncation testing requires recalculation of the CI halfwidth based on increasing amounts of truncated initial data. For a well-behaved function with some initial transients, as the number of points truncated increases, CI halfwidth will generally decrease for a time (probabilistically), then begin to rise. The point at which CI is minimized (or confidence peaks) is the optimal truncation point. This period of calculation can be computationally burdensome if the analyst does not have some prescience regarding the expected behavior of the process. Generally, the analyst will have an idea of approximately where the data function should begin to level off. Testing for truncation in such a situation requires a few calculations of confidence for truncation above, below, and at the estimated point to confirm and refine the estimate. If, however, the analyst has no idea where steady state might begin, he must recalculate confidence for truncations from the beginning and test until the confidence peak point is found. Obviously, testing for truncation of *every* point from the beginning is not necessary, but enough points must be tested to give a reasonable indication of the behavior of the confidence function, in order to find the peak. Assuming that the sample is statistically large, experience shows (as will be seen in Chapter Four) that testing for truncation at every 10 points gives good results, but to ease the amount of computation, testing for truncation at every 50 points is acceptable. Clearly, the choice depends upon both the size of the sample and the need for precision and saving of data.

It is, of course, possible that a peak in the confidence function may not exist for the given data sample. For example, if the data immediately converges to steady state with no transient period, truncation of initial points will cause the confidence to continually decrease because, while variance is not decreasing, sample size is. On the other hand, the data may never reach steady state within the given sample; it may be entirely transient. Truncation of initial points in this case will *probably* also cause the confidence to continually decrease because of the decreasing sample size; however, it is harder to tell in this case because the transient nature of the data could cause variance to either increase or decrease, which will affect confidence level. Clearly, because these functions are random, no statement about the behavior of the confidence function is entirely certain, but in most cases, the expected confidence peak should be identifiable. The special cases must be noted, and further testing is necessary in order to determine whether these can be handled effectively by the CMP. This issue will be discussed in more detail in Chapter Five.

Finally, once the optimal truncation point has been found for a data set, the values for confidence level, CI halfwidth, and total required sample size given by the CMP can be used to compare the results of steady-state detection heuristics to determine their effectiveness. In order to do this, "optimal" truncation points for a particular data set must be calculated using the candidate detection heuristics. Comparisons among these can be made on the basis of confidence. Confidence levels (or intervals) are calculated for the data set (given a constant sample size) using each truncation estimate. Clearly, the detection heuristic that yields the truncation point associated with the best confidence statistics is the most effective for that model and run. Since all of these calculations are based on probabilistic functions and behavior, many different runs of a model must be tested before a statement can be made regarding "the best heuristic" for that model. Specifics of the procedure will become clearer in Chapter Four, when examples of tested data will be described.

3.3 CONFIDENCE MAXIMIZATION PROCEDURE AS A DETECTION METHOD

We have shown how the Confidence Maximization Procedure may be used as a methodology for comparison and judgment of the effectiveness of other detection heuristics. However, it is not difficult to see that the confidence maximization concept has the potential for application as a detection method on its own. After all, the initial steps of the testing procedure involve estimation of the steady-state onset point for a complete sample using the confidence maximization concept. The only difficulty in operationalizing this set of steps for use as a detection method is in determining a way to make it "real-time". Because of our underlying interest in automating the steady-state detection process, we have no use for a method that requires that the entire data sample be collected before detection analysis can be done. Therefore, the confidence maximization concept must be made into a dynamic process.

There are many ways to go about this. One possibility would be to use a sequential method similar to Law and Kelton's (1982) sample size determination method. A general procedure of this sort would require the user to:

1) Execute the simulation by small sequential sample subsets.

2) Locate the optimal truncation point and its associated confidence each time a new subset of data is appended.

3) Stop when a pre-specified confidence requirement is met.

This sort of procedure has an additional advantage over other "real-time" detection heuristics in that not only is the optimal truncation point identified, but also the total sample size requirement is automatically determined, based on the user's precision specifications. This procedure is also nearly identical to Heidelberger and Welch's run length control procedure that was outlined in Chapter Two. The CMP can be easily transformed into a similar dynamic run length control procedure using the same basic structure. We will refer to the steady-state detection procedure that is based on the CMP as the Confidence

Maximization Rule (CMR). The specific methodology of the CMR is described in more detail in Chapter Four.

3.4 CHARACTERISTICS OF THE CONFIDENCE MAXIMIZATION RULE

On a high level, Heidelberger and Welch's approach to run length control is very much the same as the CMR; the most significant difference between these is the manner in which initial transient detection and truncation is performed. Heidelberger and Welch's transient detection method is based on Schruben's fairly complex method, and therefore it requires a substantial amount of computation at each truncation test point.

Although the CMR uses a less stringent standard to identify the existence of initial transients, the output of the CMR is effectively the same as that of Heidelberger and Welch's procedure: namely, a sample whose initial transients have been truncated so as to optimize the confidence of sample statistics. The question that remains is whether or not Schruben's transient detection method yields a significantly more precise truncation value to balance the excessive complexity and computation required. Logic indicates that the answer is no. If the objective is to maximize confidence and the basis of the procedure is the observation of the confidence function (as is true for both the CMR and Heidelberger and Welch's procedure), the addition of another mechanism to check for transients is redundant. Although this question should be studied at a future time through experiments comparing the two procedures, it seems safe to assume that the CMR will not yield results that are substantially inferior to the results of Heidelberger and Welch's procedure.

In addition to the computational savings associated with the transient detection method, the CMR requires fewer computations than Heidelberger and Welch's procedure because there is no need for batching of observations during the truncation testing period. Because of the high autocorrelation associated with single-run statistics, the variances and halfwidths calculated during the testing phase of the CMR are not valid as absolute statistics; however, the relative differences between these statistics as truncation progresses are the values of interest. These should be unaffected by the autocorrelation. Once the optimal truncation point is determined, batching can be performed to eliminate autocorrelation effects before absolute statistics are calculated.

Another reason, in addition to the computational savings, that batching is not used for the CMR truncation testing is that batching disrupts the effects of the variance reduction that occurs when initial transients are truncated. There are two ways that truncation can progress when batching is done: either individual observations may be truncated before batching, or batching may occur first and truncation is then performed on full batches of data. Each alternative has a detrimental effect on the behavior of the variance and CI halfwidth.

The first truncation alternative, batching after truncation, causes the behavior of the statistics (variance and CI halfwidth) to be misleading. The mechanics of this method are as follows: a set of observations (10, 50, 100, etc.) are truncated from the data set; the remaining data is batched into potentially large sets (possibly on the order of 1000 observations); and the change in statistics is determined. The problem is that the effect of truncating individual observations is undermined by the effects of batching on the calculation of statistics.

For example, suppose a batch size of 500 is used for a set of output data with a total of 10,000 observations, and truncations progress in groups of ten observations. Assume an initial transient exists within the first 1000 observations of the data set. The expected behavior of the confidence function is that for the first 100 groups of ten observations, confidence should increase rather smoothly; after the 1000th observation has been truncated, confidence should begin to decrease. Consider the effect of batching on these confidence calculations. Fifty groups of (ten) observations make up one batch size; therefore, as the first fifty sets of observations are truncated, transients will be eliminated, thereby decreasing the variability of the data. However, the number of batches used to determine the variance and confidence values *will remain the same throughout these* *truncations*. From the first set of ten observations truncated to the 49th set, the number of batches remaining for calculation of statistics is nineteen ([10,000/500] - 1). Hence, the variability in the data will be eliminated without any change in the sample size. Even if the data were in steady state during this period, chances are the confidence would continue to increase (or CI halfwidth to decrease) because sample size is not decreasing. This effect causes the results to be misleading.

As truncations progress beyond the batch size (in our example, beyond the 49th set of ten observations), the next complication occurs. With one additional truncation (the 50th set), the number of batches in the sample suddenly drops by one. This event will have a much more noticeable effect on the statistics calculations because the sample size was unchanged during all previous truncations. This artificially heightened effect of truncating one presumably insignificant set of observations is also misleading.

The second alternative for batching during truncation testing is to batch all observations before truncation begins. With this method, truncation may only be done for full batches of data at a time. The obvious drawback to this method is that data will often be wasted. Imagine, using our previous example, that the transient period of the data set falls at approximately the 600th (individual) observation. Assuming once again a batch size of 500, truncation of one batch will not eliminate enough of the bias; therefore, two batches, or 1000 observations, must be truncated. This means wasting 400 steady-state observations. Although the amount of data is not always a constraint to the simulationist, wasted data can sometimes be a serious drawback.

Given the drawbacks associated with the two alternatives for batching during truncation testing, batching will not be used as part of truncation testing for the CMR. While there does not appear to be any problem with using individual observations to study relative effects of truncation on the confidence function, this is another situation that should be studied further at another time.

Aside from the method of transient detection and the use of batching in determining optimal truncation point and run length, the CMR and Heidelberger and Welch's procedure are very similar. The tradeoff between the two methodologies is potentially increased precision of results (from Heidelberger and Welch's procedure) versus decreased computational intensity and increased simplicity (from the CMR). Clearly, the needs of the user will have to dictate the appropriateness of the use of the CMR over some other transient detection and/or run length control procedure such as Heidelberger and Welch's.

3.5 CONCLUSIONS

As has been shown, the Confidence Maximization Rule is one approach to operationalizing the confidence maximization principle as a steady-state detection heuristic. The Confidence Maximization Procedure, likewise, is a way to apply the principle to compare and evaluate other detection heuristics. Chapter Four will go into further detail regarding the application of these ideas.

CHAPTER FOUR
TEST AND EVALUATION

4.1 OVERVIEW OF TEST PROCEDURES

The purpose of this chapter is twofold: first, to test the effectiveness of the Confidence Maximization Rule as a steady-state detection heuristic, and second, to illustrate the use of the Confidence Maximization Procedure as a methodology for testing the effectiveness of existing detection heuristics. In order to do both, two sets of experiments were created to test these respective applications of confidence maximization.

The first set of tests evaluates the reliability and consistency of the CMR as a steady-state detection heuristic. "Reliability" refers to how sure one can be that the CMR gives a reasonable estimate of the optimal truncation point. This can be tested by checking confidence interval coverage of the theoretical mean (if known) or "large sample" mean (if theoretical mean is unknown) for several runs of a variety of models. The "consistency" of the CMR indicates how well it repeats its results for the same data set with varied sample sizes.

The second set of tests uses CI coverage to confirm that confidence maximization can reasonably be used to compare the effectiveness of various detection heuristics. These tests will illustrate how the CMP is used as a methodology for comparing detection heuristics against each other. These will also help to compare and contrast the performance of the CMR with that of the other detection heuristics tested here.

All tests conducted during this research were based on output files from multiple runs of five different models. These particular models were chosen so as to cover a reasonable spectrum of general simulation types that an average user might encounter. The five models are:

1. A simple M/M/1 queue system with a traffic intensity (ρ) arbitrarily selected to be 0.9. This model starts out "empty and idle"; therefore, it theoretically has no initialization bias/transient data region because of the regenerative nature of the system. The theoretical mean number in system for this model is nine. This model will be called "the Empty Queue Model".

2. A simple M/M/1 queue system ($\rho = 0.9$). This model starts out with 100 entities in the system (in other words, 100 people in queue when it begins); therefore, it has a positively biased transient region that exists until the natural system operation clears up the bottleneck. Given that this model has a theoretical mean number in system of nine entities, a front-loading of 100 is very powerful. This model will be called "the Loaded Queue Model".

3. A series of 15 finite-capacity M/M/1/15 queues ($\rho = 0.9$). The system starts out "empty and idle". The input to the later queues in the system is the output of the initial queues and is, therefore, affected (with a delay) by bottlenecks and other events associated with the initial queues. The issue of interest here is whether or not initialization bias will occur in the final queue as a result of the effects of the other queues on the flow. This model will be called "the Queue Series Model".

4. A simple network of two parallel capacity-constrained M/M/1/30 queues. This model begins "empty and idle", and system times build because $\rho > 1$, until queue capacities are filled, at which time balking causes the operation to stabilize. This will have a negatively biased initial transient period and, therefore, will be called "the Filling Queue Model".

5. A simple network system with no resource or queue capacity limits, whose service rate is so high that steady state is theoretically impossible to achieve. Here, the entire set of output data behaves in a transient manner. This will be called "the Transient Queue Model".

The SIMAN algorithms for these models can be found in Appendix A. For the first two models, the output variable of interest was total number in system; for the Filling Queue and Transient Queue models, the variable of interest was the total number in the second "station" or queue subsystem. Finally, for the Queue Series model, the variable of interest was the number in the subsystem at the final (fifteenth) queue.

4.2 CONFIDENCE MAXIMIZATION AS A DETECTION HEURISTIC

4.2.1 Reliability

The first test run on the CMR was an evaluation of its reliability as a detection heuristic; in other words, how sure can one be that the truncation point chosen by the CMR is actually optimal and will produce statistics with as high a confidence as possible? Confidence in a statistic is partially a matter of confidence level and interval size, but another important aspect is the "correctness" of the confidence interval. This can be measured for samples with a known theoretical mean by checking the coverage of sample mean confidence intervals. A "good" confidence interval should result in coverage of the theoretical mean (1-α) percent of the time (where (1-α) is the originally assigned confidence level). For samples with an unknown theoretical mean, an estimate can be computed by taking statistics on samples of significantly larger size than the sample of interest. Coverage for such samples can be checked by determining whether or not the sample confidence interval comes within a (100*(1-α)) percent CI halfwidth of a Normal distribution around the large sample statistic. A logical way to compute the size of this halfwidth is to run several large samples and calculate the (100*(1-α)) percent statistics for the means of the large samples.

Testing of the reliability began with ten different runs of each of the five test models. Based on computer software and hardware constraints associated with this project, each run consisted of 5000 observations. A Fortran program called "Statistics" (listed in Appendix B) was developed in order to compute and compile the mean, standard deviation, and 95 percent CI halfwidth, truncating initial points in groups of ten. In other words, the output of this program is 500 lines of statistics, the first line giving statistics based on zero points truncated, the second line on ten points truncated, and so on, until the 500th line gives statistics based on just the last ten points of the sample. The statistics computed in this program are based on unbatched samples, for reasons that were given in Chapter Three. The truncation statistics listing associated with each run was then used to determine the associated CMR truncation point by identifying the point at which the CI halfwidth is minimized.

The total sample sizes (including both truncated and untruncated data) used for these calculations were held constant at 5000; no additional observations were added to the ends of the samples to "make up" for truncated initial observations. In other words, the sizes of the samples actually used for the calculations varied depending on the number of points truncated. The reason for this approach is that with truncation of initial transients, there is an implicit tradeoff between confidence and sample size. In general, if a heuristic removes observations, whether or not these are part of the steady state, confidence will, at least, stay fairly constant and, at best, improve if new observations are allowed to be added at the end of the sample to replace the eliminated initial observations. The cost of such an increase in confidence is that sample size requirements increase. By assigning a constant total sample size for these tests, confidence increases and decreases that result from transient elimination occur without affecting the sample size requirement. Shown this way, an increase in confidence is strictly a benefit to the user because there is no associated cost of increased sample size that must be taken into account. In most cases, once the optimal truncation point has been identified, any increases in sample size will only further increase (probabilistically) the confidence of the statistics.

To give the reader a feel for the effect of CMR truncation on output data, the output of one run from each model is shown in Figures 4.1 through 4.5. Because of graphical constraints, the data has been batched into groups of 25 observations; however, the data trends of interest here are unaffected by this batching. The data to be truncated by the CMR is represented by squares with dots inside, while the untruncated data is represented by filled-in squares. These two groups together make up the full set of 5000 observations (25 observations/batch * 200 batches). Note from these graphs that no truncation is required for the Empty Queue and Queue Series models (Figures 4.1 and 4.3) because the initial conditions are within "normal steady-state range". However, the CMR removes the

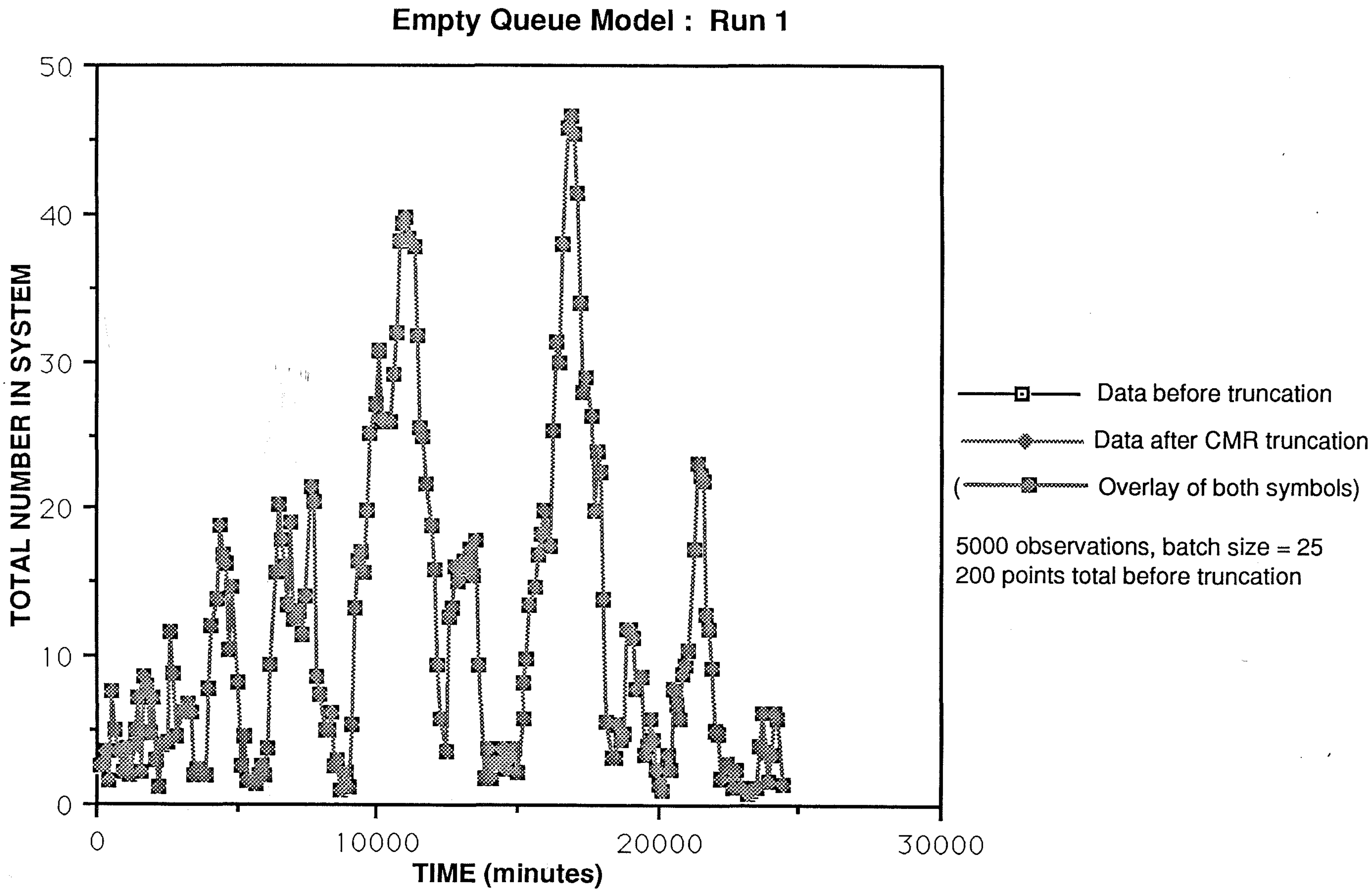
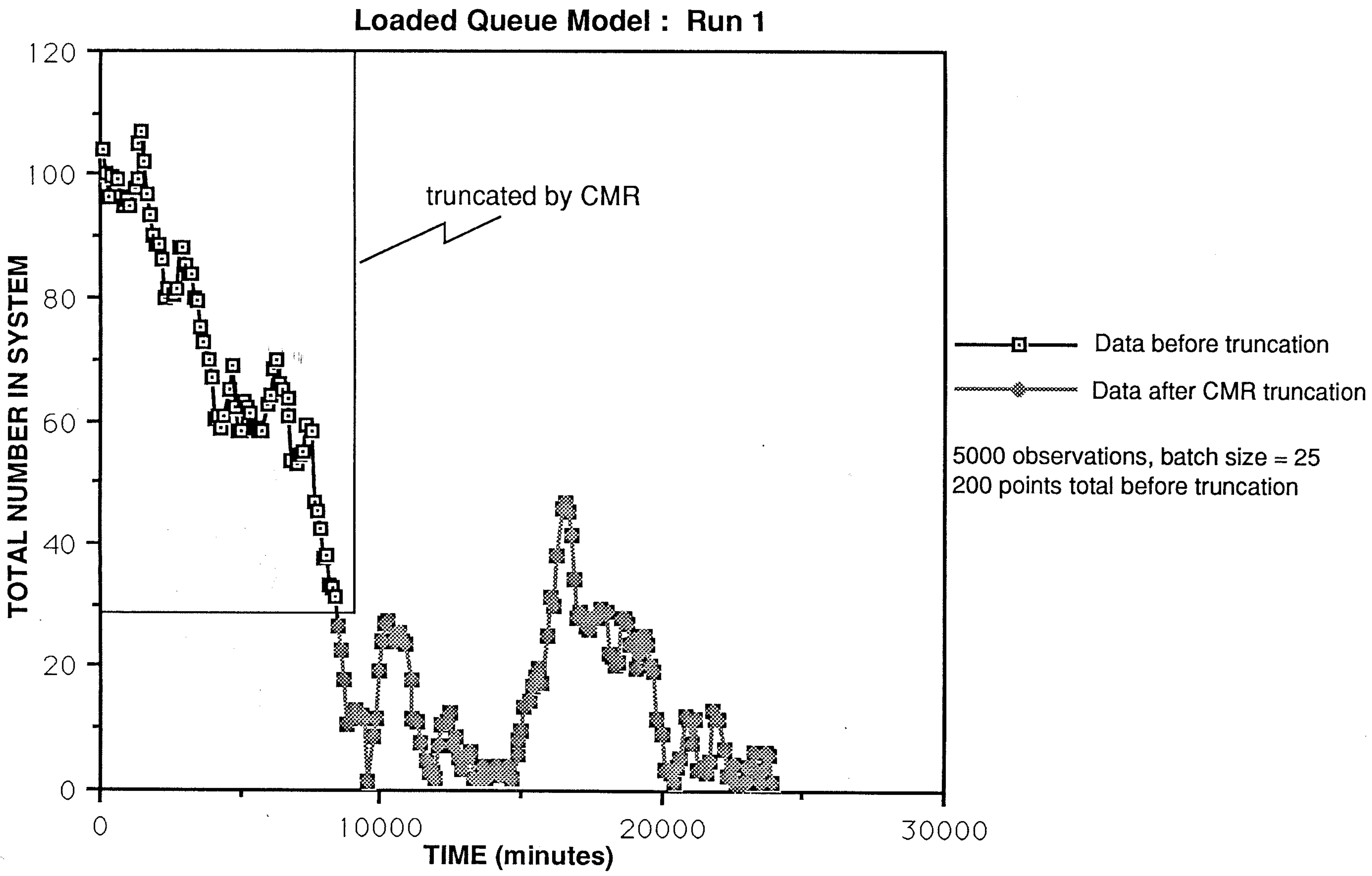
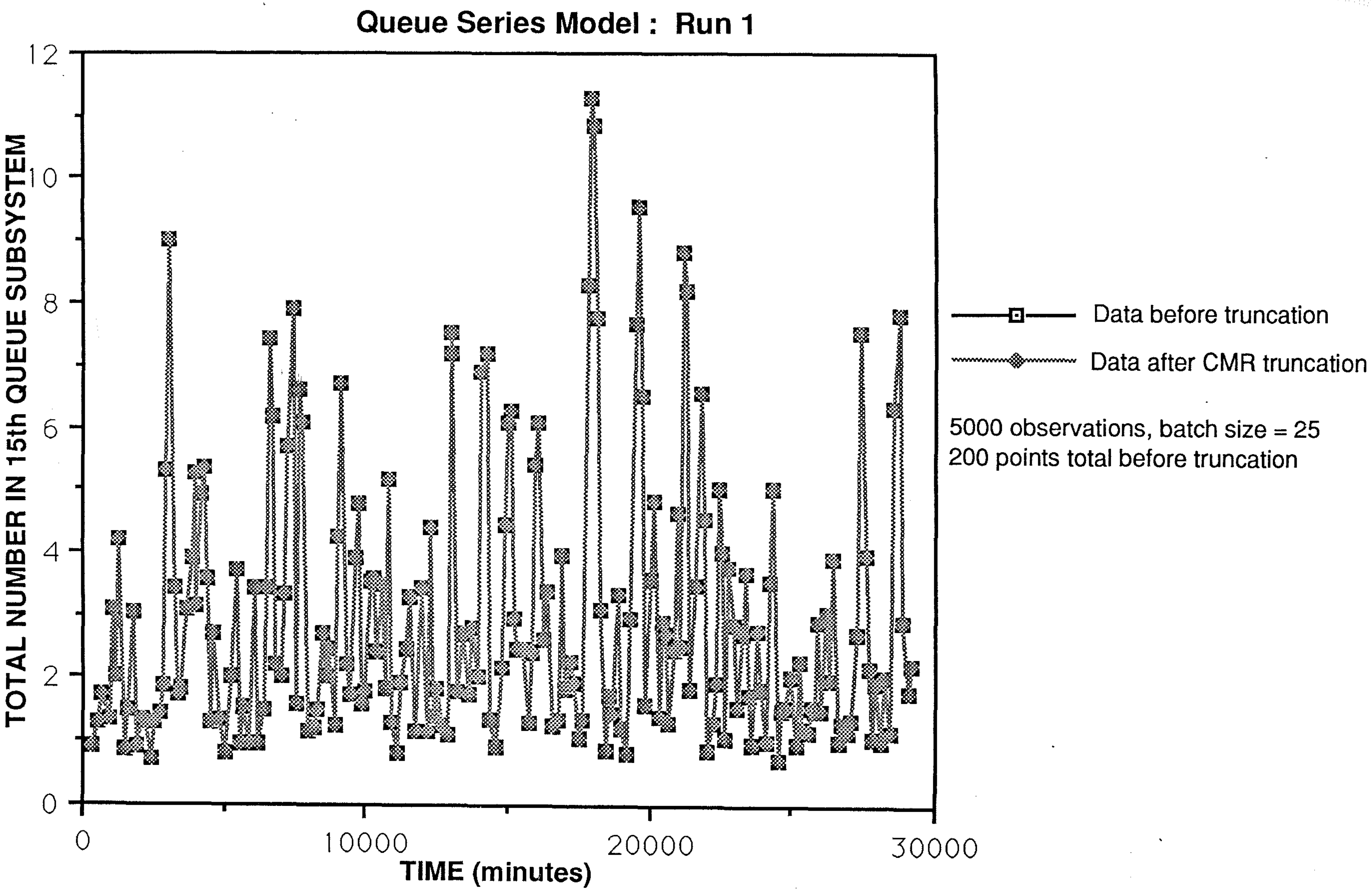
Empty Queue Model : Run 1
50
40
30
20
10
0
TOTAL NUMBER IN SYSTEM
0
10000
20000
30000
TIME (minutes)
Data before truncation
Data after CMR truncation
(Overlay of both symbols)
5000 observations, batch size = 25
200 points total before truncation

Figure 4.1



5000 observations, batch size = 25
200 points total before truncation

Figure 4.2



5000 observations, batch size = 25
200 points total before truncation

Figure 4.3

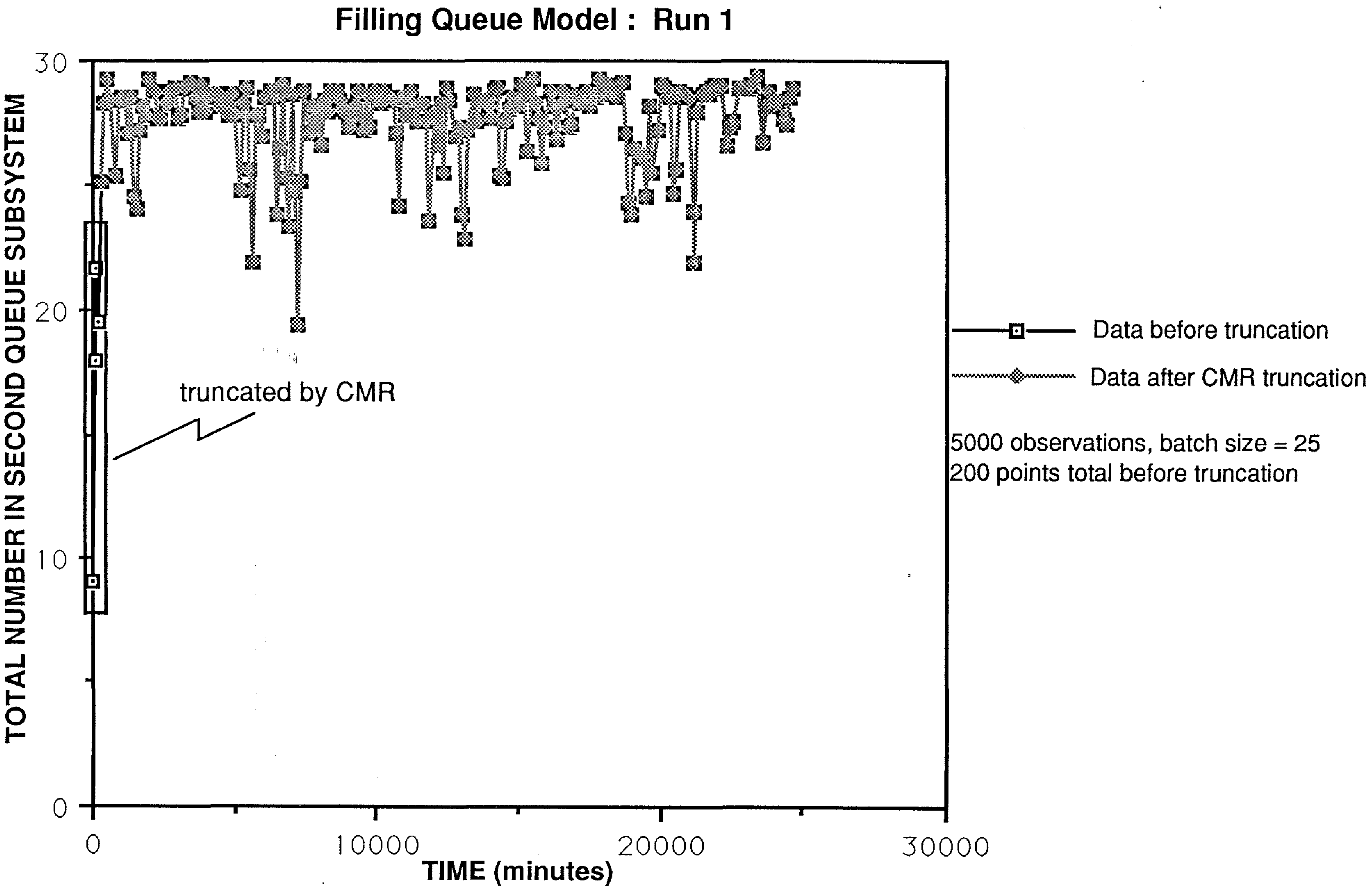

Filling Queue Model : Run 1
TOTAL NUMBER IN SECOND QUEUE SUBSYSTEM
TIME (minutes)
30
20
10
0
0
10000
20000
30000
truncated by CMR
Data before truncation
Data after CMR truncation
5000 observations, batch size = 25
200 points total before truncation

Figure 4.4

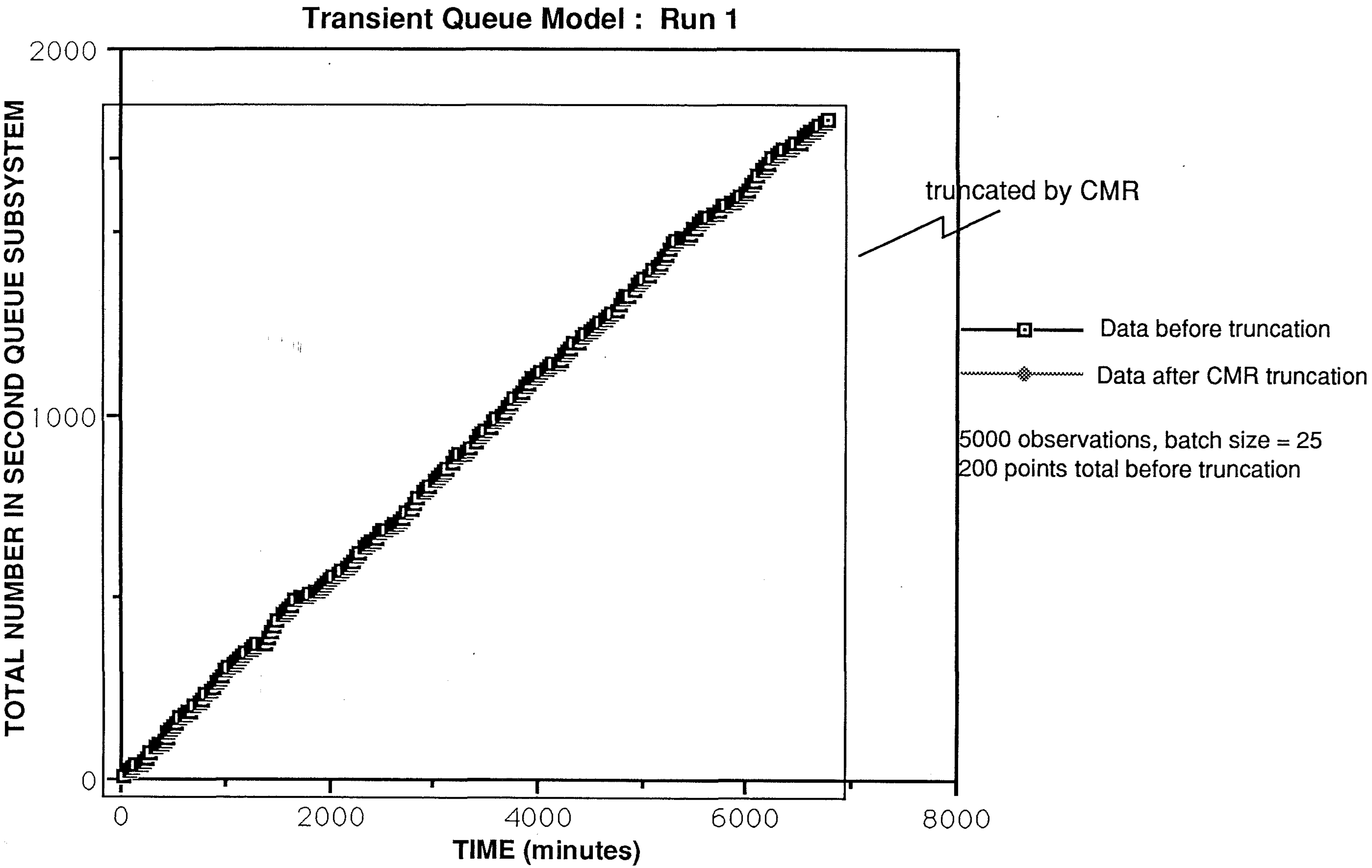

Transient Queue Model : Run 1
2000
1000
0
TOTAL NUMBER IN SECOND QUEUE SUBSYSTEM
0
2000
4000
6000
8000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
5000 observations, batch size = 25
200 points total before truncation

Figure 4.5

positively biased initial data from the Loaded Queue Model (Figure 4.2) and the negatively biased initial data from the Filling Queue Model (Figure 4.4) as appropriate. The most unique output set shown here is the Transient Queue Model (Figure 4.5), in which the CMR cannot identify an appropriate truncation point and, therefore, indicates that the whole set of data should be truncated. Since it is known theoretically that there is no steady state for this model, the CMR is correct. Similar graphs for the nine remaining runs of the five models can be found in Appendix C. Tables 4.1, 4.3, 4.5, 4.7, and 4.9 present the statistics associated with the (unbatched) output data following CMR truncation for the runs of the Empty Queue Model, the Loaded Queue Model, the Queue Series Model, the Filling Queue Model, and the Transient Queue Model, respectively.

The statistics and graphs associated with the last two runs for the Loaded Queue Model should especially be noted. These output sets exhibit the somewhat unusual characteristic of stabilizing for a fairly long period of time at the initial, overloaded level of operation (see graphs in Appendix C, pages C-18 and C-19 for illustration of this behavior). The result of this occurrence is that the CMR sees the high level as the steady-state level because there are not enough observations at the lower, actual steady-state level to cause a significant change in the system statistics as initial observations are truncated. The CMR, therefore, indicates that no observations should be truncated. This is clearly a fallacy and will be discussed in more depth in the "Consistency" section of this chapter.

The first step in evaluating the reliability of the CMR required estimating the CI coverages associated with each model. In order to determine CI coverage effectively, non-terminating simulation output must be batched or in some other way massaged to remove the effects of autocorrelation. The sample size restriction of 5000 observations dictated a limit of 100 points/batch as the largest reasonable batch size; this resulted in a base sample size (before truncation) of fifty batches from which to take statistics. Because of the inconsistencies in statistics that are based on small non-Normal samples, and because the distributions of these output sets were not all known, a conservative value of fifty was

Empty Queue Model Statistics
CMR Truncation

Table 4.1
Unbatched Data

MODEL: EMPTY Q ACTUAL MEAN: 9.0

SAMPLE SIZE: 5000 UNBATCHED

RUN	POINTS TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	% TRUNC.
1	0	11.75	10.78	0.299	0
2	0	9.27	8.22	0.228	0
3	0	7.45	5.49	0.152	0
4	0	6.71	7.36	0.204	0
5	0	8.03	7.50	0.208	0
6	0	6.29	5.38	0.149	0
7	0	7.25	7.00	0.194	0
8	0	5.84	5.06	0.140	0
9	0	20.73	21.33	0.591	0
10	0	14.67	17.87	0.495	0

Table 4.2
Batched Data

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	COVERAGE	% TRUNC.	# BATCHES
1	0	11.75	10.30	3.00	+	0	50
2	0	9.27	7.39	2.15	+	0	50
3	0	7.45	4.36	1.27		0	50
4	0	6.71	6.76	1.97		0	50
5	0	8.03	6.90	2.01	+	0	50
6	0	6.29	4.26	1.24		0	50
7	0	7.25	6.31	1.83	+	0	50
8	0	5.84	3.93	1.15		0	50
9	0	20.73	21.20	6.18		0	50
10	0	14.67	17.70	5.14		0	50
				COVERAGE:	40%		

CIHW: confidence interval halfwidth

Loaded Queue Model Statistics
CMR Truncation

Table 4.3
Unbatched Data

MODEL: LOADEDQ ACTUAL MEAN: 9.0

SAMPLE SIZE: 5000 UNBATCHED

RUN	POINTS TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	% TRUNC.
1	1820	14.38	11.29	0.393	0.364
2	1180	11.06	8.75	0.278	0.236
3	1260	9.65	7.57	0.243	0.252
4	970	13.40	13.53	0.418	0.194
5	1610	12.62	8.74	0.294	0.322
6	1230	6.38	4.95	0.158	0.246
7	2960	5.02	4.03	0.175	0.592
8	2510	6.83	5.02	0.197	0.502
9	0	54.91	33.75	0.936	0
10	0	70.20	41.60	1.150	0

Table 4.4
Batched Data

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	COVERAGE	% TRUNC.	# BATCHES
1	18	14.38	10.90	4.09		0.36	32
2	12	11.06	7.86	2.66	+	0.24	38
3	13	9.65	6.67	2.29	+	0.26	37
4	10	13.40	13.20	4.40	+	0.20	40
5	16	12.62	8.26	3.00		0.32	34
6	12	6.38	3.78	1.36		0.24	38
7	30	5.02	2.89	1.44		0.60	20
8	25	6.83	3.66	1.59		0.50	25
9	0	54.91	33.90	9.63		0.00	50
10	0	70.20	41.80	11.90		0.00	50
				COVERAGE:	30%		

CIHW: confidence interval halfwidth

Queue Series Model Statistics
CMR Truncation

Table 4.5
Unbatched Data

MODEL: SERIES Q THEORET. MEAN: 5.36

SAMPLE SIZE: 5000 UNBATCHED

RUN	POINTS TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	% TRUNC.
1	0	3.02	2.69	0.075	0
2	0	3.91	3.23	0.090	0
3	0	3.80	3.35	0.093	0
4	0	3.51	2.92	0.081	0
5	0	3.50	3.09	0.086	0
6	0	3.46	2.83	0.078	0
7	0	3.72	3.30	0.091	0
8	0	3.81	3.31	0.092	0
9	0	3.45	3.10	0.086	0
10	0	3.38	3.04	0.084	0

CIHW: confidence interval halfwidth

Table 4.6
Batched Data

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	EXTENDED RUN MEAN	COV. (EXT. RUN MEAN)	% TRUNC.	#BATCHES USED
1	0	3.02	1.24	0.360	3.34	0%	0	50
2	0	3.91	1.90	0.554	3.83	100%	0	50
3	0	3.80	2.25	0.657	3.89	100%	0	50
4	0	3.51	1.58	0.460	3.79	100%	0	50
5	0	3.50	2.20	0.642	3.47	100%	0	50
6	0	3.46	1.75	0.511	3.70	100%	0	50
7	0	3.72	2.33	0.680	3.53	100%	0	50
8	0	3.81	2.12	0.620	3.72	100%	0	50
9	0	3.45	2.13	0.621	3.62	100%	0	50
10	0	3.38	1.78	0.519	3.44	100%	0	50
					-----------	----------		
		EXT. RUN MEAN:			3.633	90%		
		EXT. RUN HW:			0.113			

Filling Queue Model Statistics
CMR Truncation

Table 4.7
Unbatched Data

MODEL: FILLING Q THEORET. MEAN: unknown

SAMPLE SIZE: 5000 UNBATCHED

RUN	POINTS TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	% TRUNC.
1	100	27.65	2.17	0.061	0.02
2	130	27.48	2.54	0.071	0.026
3	150	27.69	2.28	0.064	0.03
4	140	27.72	2.14	0.060	0.028
5	90	27.67	2.16	0.060	0.018
6	100	27.33	2.55	0.071	0.02
7	110	27.67	2.05	0.058	0.022
8	110	27.11	2.89	0.081	0.022
9	210	27.73	2.15	0.061	0.042
10	130	27.30	2.58	0.072	0.026

CIHW: confidence interval halfwidth

Table 4.8
Batched Data

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	EXTENDED RUN MEAN	COV. (EXT. RUN MEAN)	% TRUNC.	#BATCHES USED
1	1	27.65	1.01	0.291	27.50	100%	0.02	49
2	1	27.48	1.48	0.425	27.50	100%	0.02	49
3	2	27.69	1.21	0.351	27.60	100%	0.04	48
4	1	27.72	1.04	0.299	27.50	100%	0.02	49
5	1	27.67	1.18	0.338	27.60	100%	0.02	49
6	1	27.33	1.46	0.420	27.50	100%	0.02	49
7	1	27.67	1.07	0.309	27.50	100%	0.02	49
8	1	27.11	2.08	0.597	27.20	100%	0.02	49
9	2	27.73	1.21	0.350	27.40	100%	0.04	48
10	1	27.30	1.33	0.383	27.50	100%	0.02	49
			EXT. RUN MEAN:		27.48	100%		
			EXT. RUN HW:		0.071			

Transient Queue Model Statistics
CMR Truncation

Table 4.9
Unbatched Data

MODEL: TRANS Q THEORET. MEAN: unknown

SAMPLE SIZE: 5000 UNBATCHED

RUN	POINTS TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	% TRUNC.
1	all	n/a	n/a	n/a	100%
2	all	n/a	n/a	n/a	100%
3	all	n/a	n/a	n/a	100%
4	all	n/a	n/a	n/a	100%
5	all	n/a	n/a	n/a	100%
6	all	n/a	n/a	n/a	100%
7	all	n/a	n/a	n/a	100%
8	all	n/a	n/a	n/a	100%
9	all	n/a	n/a	n/a	100%
10	all	n/a	n/a	n/a	100%

Table 4.10
Batched Data

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED BY CMR	MEAN	STAN. DEV.	95% CIHW	EXTENDED RUN MEAN	COV. (EXT. RUN MEAN)	% TRUNC.	#BATCHES USED
1	all	n/a	n/a	n/a	n/a	n/a	100%	0
2	all	n/a	n/a	n/a	n/a	n/a	100%	0
3	all	n/a	n/a	n/a	n/a	n/a	100%	0
4	all	n/a	n/a	n/a	n/a	n/a	100%	0
5	all	n/a	n/a	n/a	n/a	n/a	100%	0
6	all	n/a	n/a	n/a	n/a	n/a	100%	0
7	all	n/a	n/a	n/a	n/a	n/a	100%	0
8	all	n/a	n/a	n/a	n/a	n/a	100%	0
9	all	n/a	n/a	n/a	n/a	n/a	100%	0
10	all	n/a	n/a	n/a	n/a	n/a	100%	0

determined to be the smallest reasonable starting sample size. The equations used to calculate the statistics for these batched runs were based on Student's t distribution, rather than Normal distribution tables, because of the reduced sample sizes involved.

The statistics associated with batched output following CMR truncation for the five models are shown in Tables 4.2, 4.4, 4.6, 4.8, and 4.10. These statistics include coverage estimates for each model, based on its ten runs. Coverage for the models with unknown theoretical means was determined in the manner described earlier in this chapter. Each model was run again using the same ten random number seeds, but this time with a sample size of 30,000 observations. The means for these "extended runs" were calculated and the distribution of these means was estimated for each model. Coverages were determined by calculating the proportion of the extended run CI halfwidth that was covered by the normal run halfwidth estimate for each run. The ten estimates were then averaged to determine the coverage for a particular model.

Unfortunately, for models with a high degree of variability, 100 observations is not generally a large enough batch size to effectively reduce the autocorrelation. Such models include all three of the M/M/1 queue models tested here: Empty Queue, Loaded Queue, and Queue Series. The effect of the remaining autocorrelation is that the size of the CI halfwidth shows up as unrealistically small; hence, the coverage computed for each of these models is much lower than expected.

For the Empty Queue Model, coverage for the test runs is 40 percent; for the Loaded Queue Model, coverage is only 30 percent. Since the theoretical "optimal" truncation point for the Empty Queue Model is known to be zero in all cases, and the CMR identifies the optimal truncation points correctly, the coverage value of 40 percent must be about the highest achievable given the sample/batch size constraints. Therefore, the coverage achieved for the Loaded Queue Model is fairly high, relative to the highest achievable coverage value.

Coverage for the Queue Series Model is 90 percent, quite a bit better than the other two. Because of the unusual input function for this model (the output of fourteen queues), normal queuing theory calculations do not apply; therefore, it was determined that using extended run statistics as the basis for comparison for the Queue Series Model would be more meaningful. The main reason for the superiority of the Queue Series Model coverage is that the queues are constrained to a maximum of fifteen entities. Limiting the number allowed in each queue causes the amount of variability in the output to decrease drastically. Coverage for the Filling Queue Model is much better than the simple queue models; this is logical because it is a model based on two parallel constrained queues, and its output is a great deal less variable than the single, unconstrained queue outputs. Coverage for the Transient Queue Model is actually meaningless for the CMR because the CMR calls for truncation of all data in these runs.

Given the sample size constraints associated with these runs, the coverages calculated for these models are not as low as they might seem. For the Queue Series Model, the Filling Queue Model, and the Transient Queue Model, the results are about as good as can be expected theoretically. The statistics from the more variable Empty Queue and Loaded Queue models are actually not far from the theoretical statistics; most of their CI halfwidths come within far less than 1.0 (approximately 10 percent) of the theoretical mean.

To show the effect of CMR truncation, Tables 4.2, 4.11, 4.6, 4.12, and 4.13 give the batched statistics of the same ten runs of the five models (respectively) for the case of no truncation. Using CMR truncation results in two significant changes to the output statistics: first, the size of the CI halfwidth decreases, indicating that the user can be more confident of the accuracy of the estimated statistics; second, the coverage of the estimated statistics, in general, improves. Clearly, for the Empty Queue and Queue Series models, for which zero is the optimal truncation point, the statistics and coverage remain the same.

Table 4.11
Loaded Queue Model Output Statistics
No Truncation
Batched Data

MODEL: LOADEDQ ACTUAL MEAN: 9.00

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNCATED	MEAN	STAN. DEV.	95% CIHW	COVERAGE	% TRUNC.	# BATCHES USED
1	0	35.50	31.70	9.000	0	0	50
2	0	23.70	25.50	7.240	0	0	50
3	0	22.40	26.10	7.420	0	0	50
4	0	26.60	30.70	8.720	0	0	50
5	0	31.40	31.30	8.910	0	0	50
6	0	16.20	21.20	6.020	0	0	50
7	0	39.60	36.20	10.300	0	0	50
8	0	38.30	38.30	10.900	0	0	50
9	0	54.90	33.90	9.630	0	0	50
10	0	70.20	41.80	11.900	0	0	50
				COVERAGE:	0%		

Table 4.12
Filling Queue Model Output Statistics
No Truncation
Batched Data

MODEL: FILLING Q THEOR. MEAN: unknown

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNC.	MEAN	STAN. DEV.	95% CIHW	EXTEN. RUN MEAN	COV. (EX.RUN MEAN)	% TRUNC.	#BATCHES USED
1	0	27.40	1.80	0.526	27.5	100%	0	50
2	0	27.10	2.58	0.753	27.5	100%	0	50
3	0	27.30	2.62	0.763	27.6	100%	0	50
4	0	27.40	1.84	0.537	27.5	100%	0	50
5	0	27.40	2.24	0.653	27.6	100%	0	50
6	0	27.10	2.40	0.698	27.5	100%	0	50
7	0	27.40	2.07	0.604	27.5	100%	0	50
8	0	26.70	3.25	0.947	27.2	100%	0	50
9	0	27.20	2.69	0.786	27.4	100%	0	50
10	0	27.00	2.14	0.624	27.5	100%	0	50
			EXT. RUN	MEAN	27.48	100%		
			EXT. RUN	HW	0.071			

Table 4.13
Transient Queue Model Output Statistics
No Truncation
Batched Data

MODEL: TRANS Q THEOR. MEAN: unknown

SAMPLE SIZE: 5000 BATCHED 100 OBS/ BATCH

RUN	BATCHES TRUNC.	MEAN	STAN. DEV.	95% CIHW	EXTEN. RUN MEAN	COV. (EX.RUN MEAN)	% TRUNC.	#BATCHES USED
1	0	922.00	537.00	153	1289.0	0%	0	50
2	0	971.00	547.00	155	1354.0	0%	0	50
3	0	917.00	536.00	152	1279.0	0%	0	50
4	0	892.00	534.00	152	1284.0	0%	0	50
5	0	959.00	554.00	157	1291.0	0%	0	50
6	0	961.00	562.00	160	1354.0	0%	0	50
7	0	917.00	547.00	155	1270.0	0%	0	50
8	0	944.00	559.00	159	1298.0	0%	0	50
9	0	902.00	498.00	142	1294.0	0%	0	50
10	0	871.00	530.00	151	1259.0	0%	0	50
					-----------	-----------		
			EXT. RUN	MEAN	1297.2	0%		
			EXT. RUN	HW	4.05			

However, for the Loaded Queue Model, the computed CI halfwidths with no truncation are a great deal larger than with CMR truncation; likewise, the coverage with no truncation is zero percent. This represents a serious degradation in the usefulness of the output statistics. Note that for the CMR truncated runs, no new observations are added to the end of the sample to offset the removal of the initial, truncated observations. What this shows is that even with a substantial decrease in the sample size from which the statistics are taken (under CMR truncation), the validity of the output is greatly enhanced by the removal of the transient data. The effect of CMR truncation on the Filling Queue Model is less notable than the case of the Loaded Queue Model. Here, the coverages are unaffected by truncation, but the CI halfwidths are significantly enlarged--nearly doubled, in most cases--by keeping the initial transients in the statistical calculations. The effect is less extreme here because of the relatively small size of the transient period as compared to the full data set; the transients are generally only about two to three percent of the output. Again, the Transient Queue Model is meaningless in this illustration because its statistics with no data truncated are terrible, and there can be no comparison with CMR truncation because the CMR can find no optimal truncation point for any of the runs.

Based on the overall CI halfwidth and coverage levels produced by CMR truncation on the runs of the five models, the CMR has shown itself to be a reasonably reliable method for transient elimination, given the sample size constraints associated with this project. The reliability of the CMR will be confirmed at a later point in this chapter, at which time the CMR results will be compared with results of other truncation methods. Clearly, the coverage issue as it relates to highly variable data such as the Empty Queue and Loaded Queue models should be studied in more depth at a later time and without the sample size constraints encountered here.

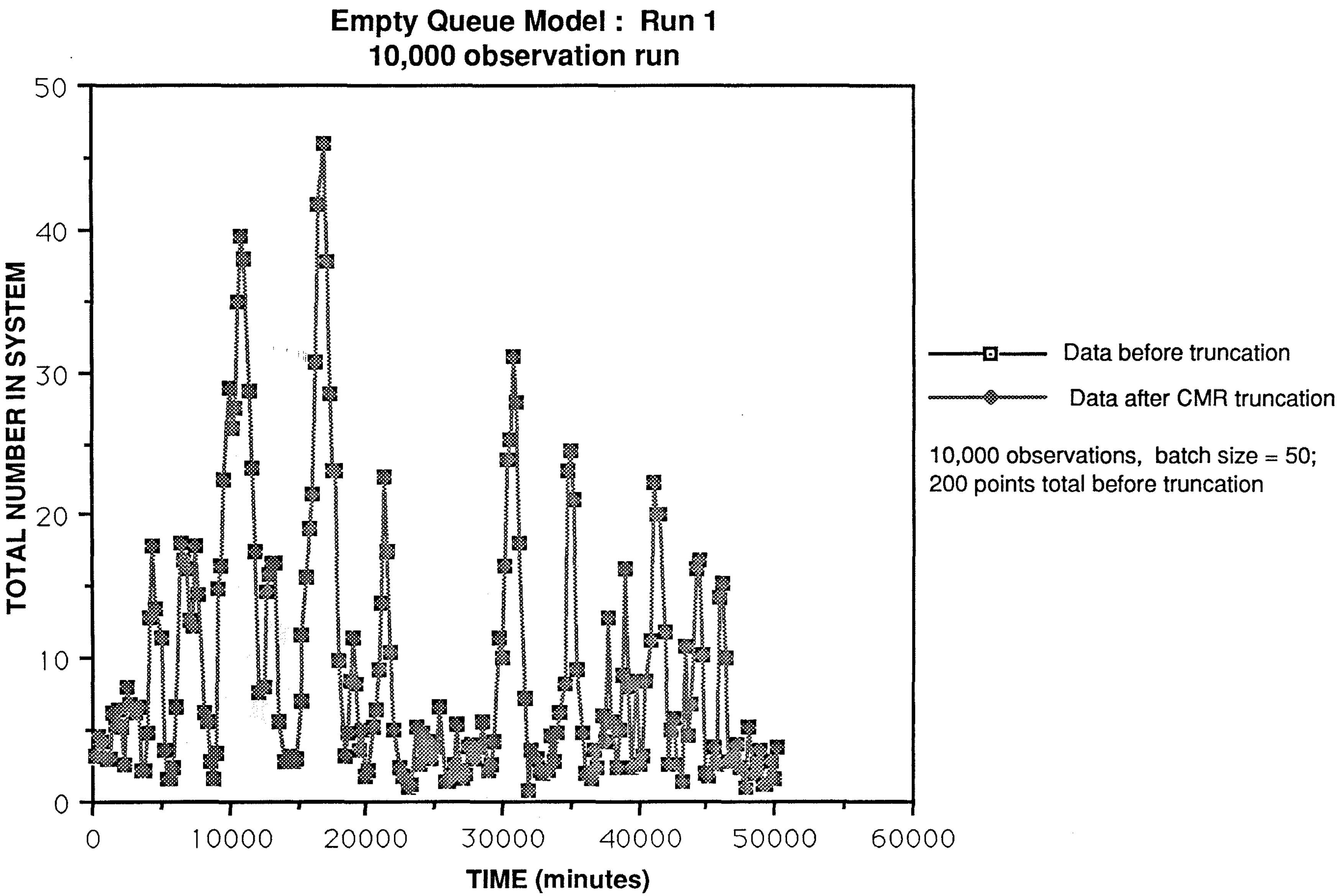
4.2.2 Consistency

The second test run on the CMR was a simple test of the consistency of its results. For this test, a second set of samples was collected for the first five of the ten runs of each of the five models, but this time 10,000 observations were collected instead of 5000. The purpose was to determine the CMR truncation point for each of these longer runs and confirm that it closely coincides with the CMR truncation point calculated for the same run of 5000 observations.

A modified version of the "Statistics" program was used on the new, longer runs. Because of the increased amount of data (hence, increased computation required), statistics were only calculated after truncation of sets of fifty points, rather than sets of ten points. The CMR truncation points for each of these, therefore, will be multiples of fifty and will not exactly coincide with the truncation points for the short runs; however, these estimates will still effectively illustrate whether or not the CMR is consistent within the runs. Figures 4.6 through 4.10 are graphs of the first 10,000 observation run of each of the five models (runs with the same initial random number streams as those shown in Figures 4.1 through 4.5).

Tables 4.14 through 4.18 list the CMR truncation points computed for the 5000 observation and 10,000 observation runs of each of the five models, along with their associated means and CI halfwidths. (Coverage values are not shown here because, in the context of unbatched output, coverage is less meaningful.) These lists of truncation points indicate that, on the whole, the CMR is very consistent in its results for each run. The only discrepancy that occurs is in the third run of the Loaded Queue Model, in which the long run CMR truncation point is 300 observations less than the short run truncation point. Out of twenty-five truncation points, one discrepancy is not unreasonable. For the most part, it can be said that when using the CMR, the increased confidence in system statistics obtained from a larger sample occurs simply because the number of observations in the calculations is larger; the larger sample size will not usually make the CMR "perform" any better.

However, this observation does not hold up for all data sets. Recall the last two runs of the Loaded Queue Model that were singled out at the beginning of this chapter because of their unusually long and seemingly "stable" transient periods, which the CMR



10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.6

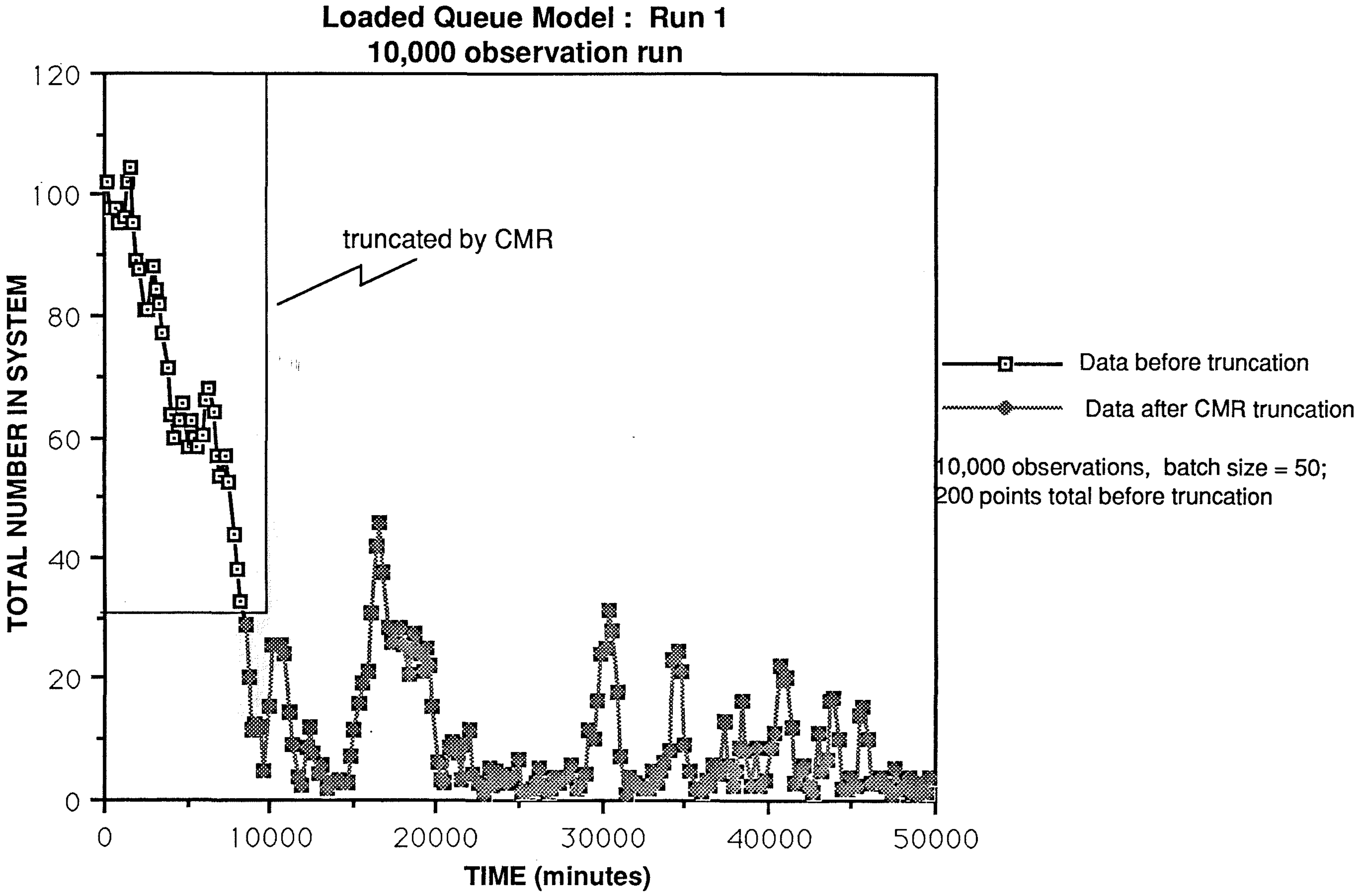
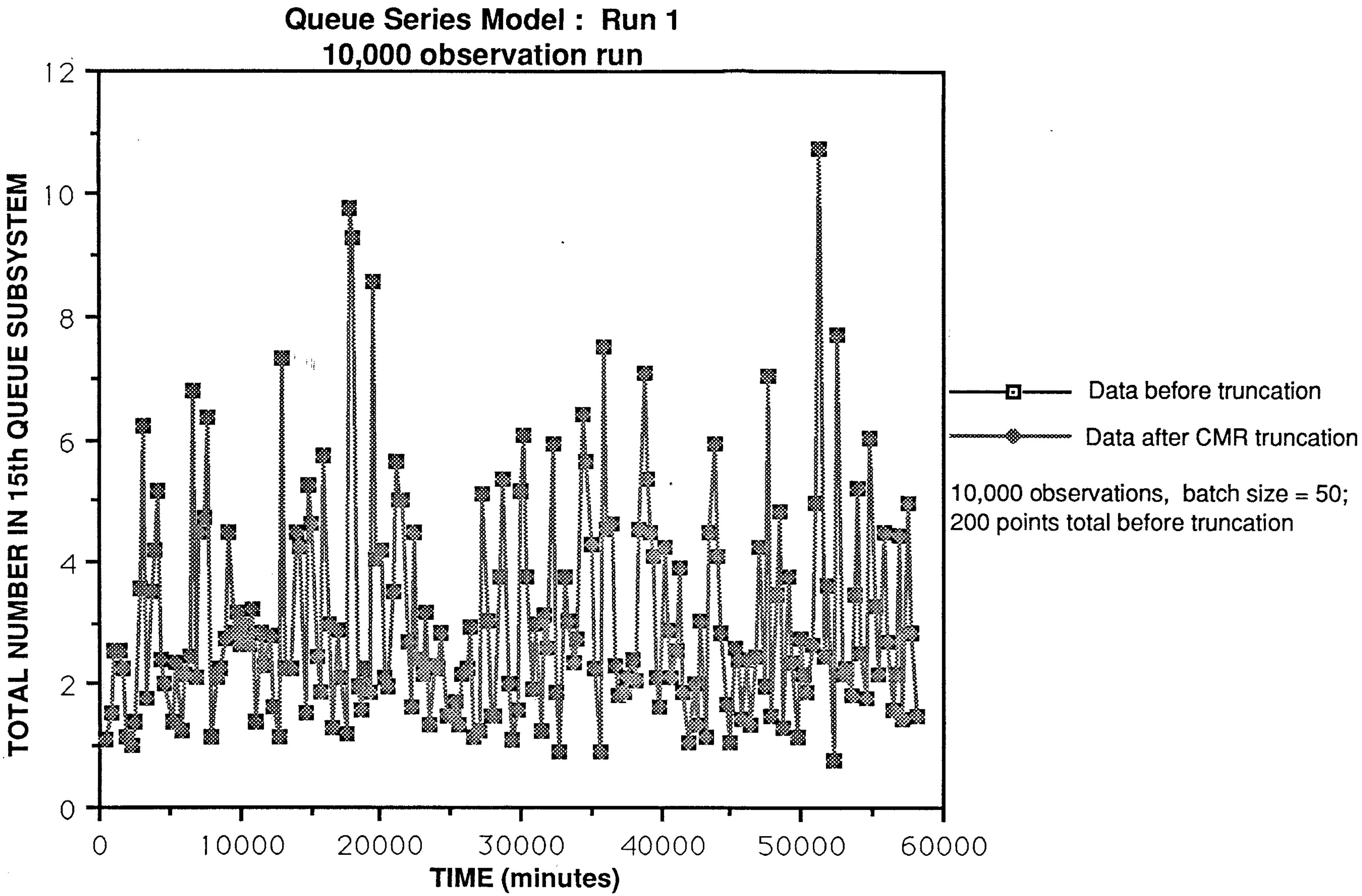

Loaded Queue Model : Run 1
10,000 observation run
120
100
80
60
40
20
0
TOTAL NUMBER IN SYSTEM
0
10000
20000
30000
40000
50000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.7


Queue Series Model : Run 1
10,000 observation run

10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.8

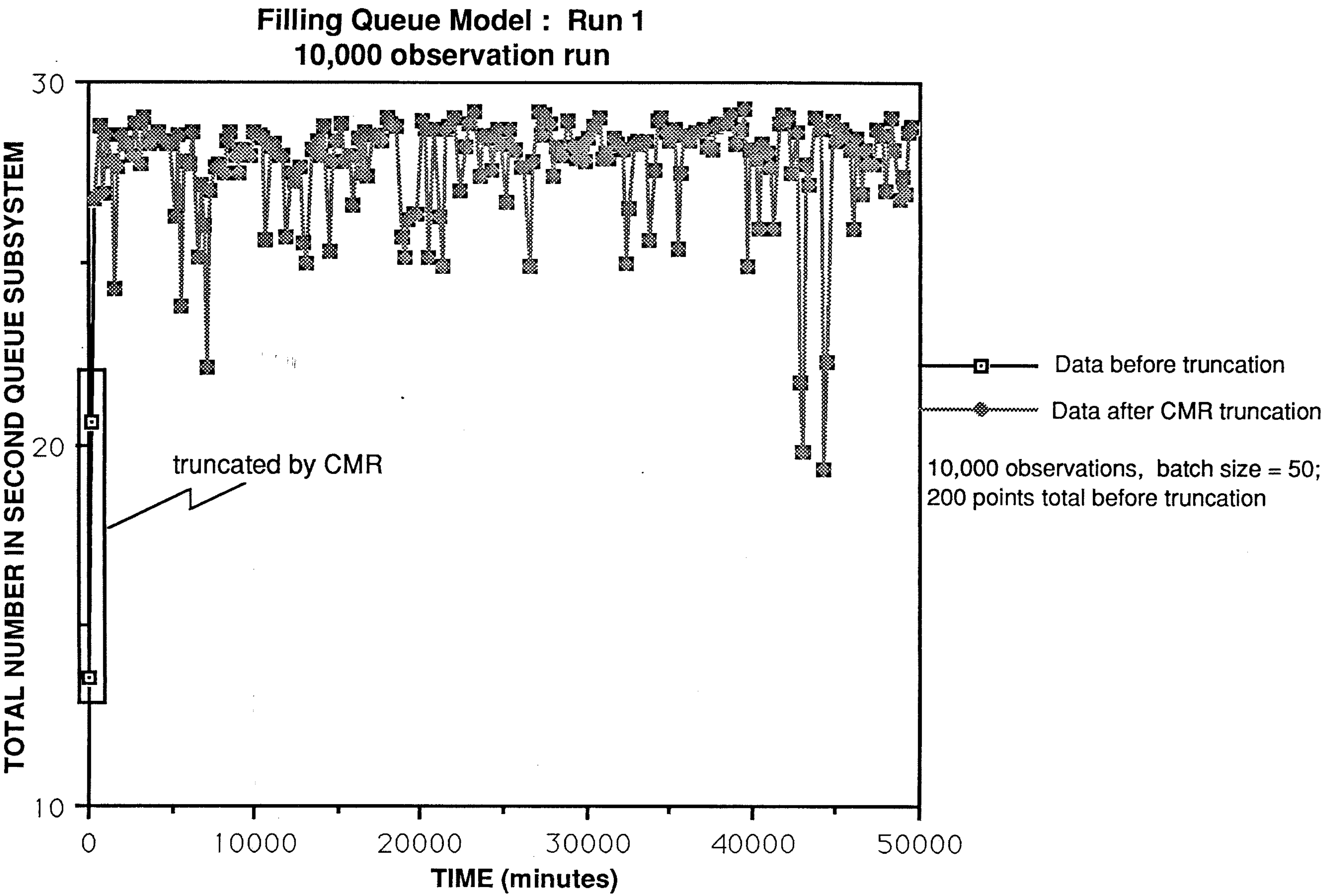
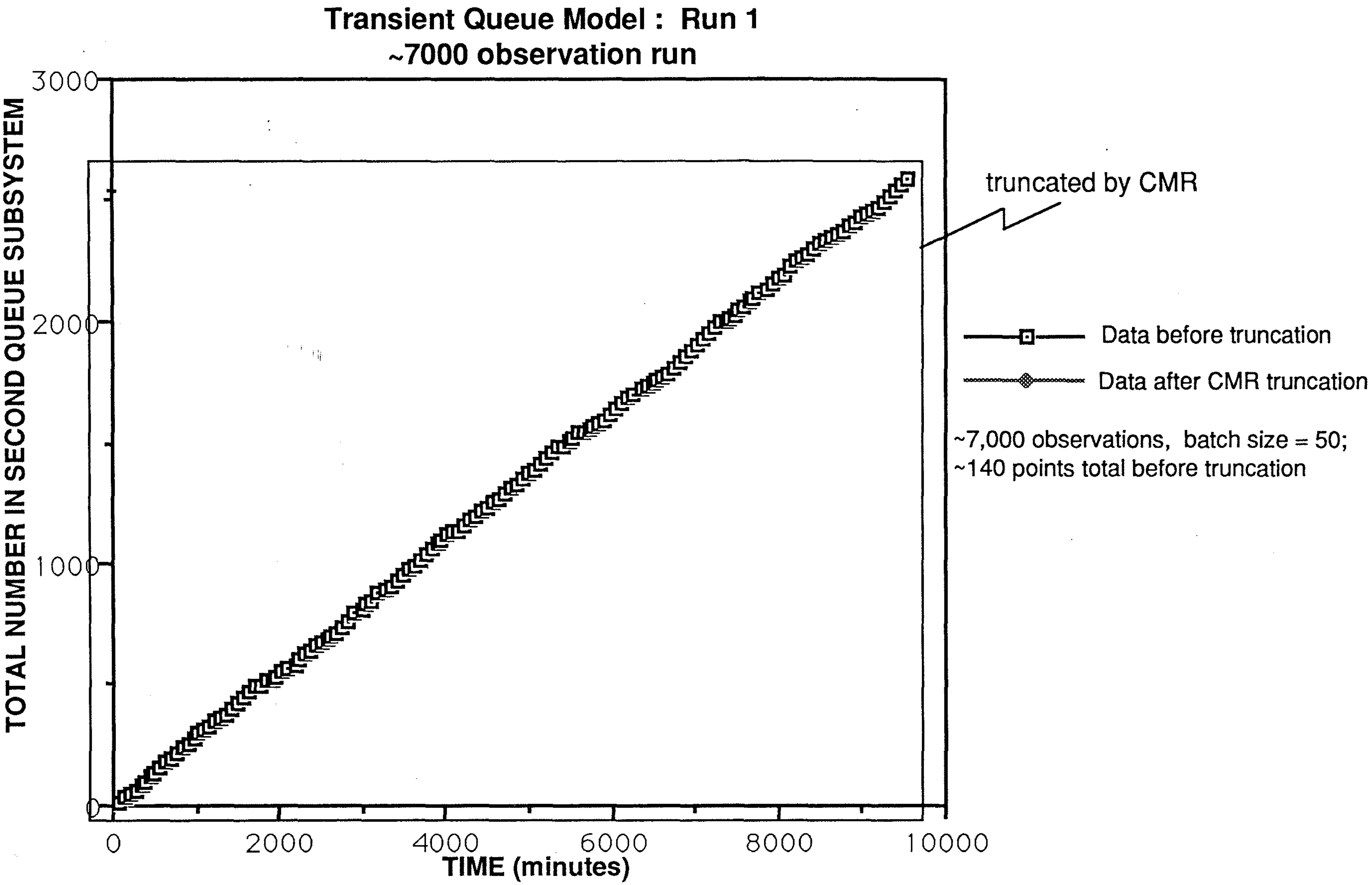

Filling Queue Model : Run 1
10,000 observation run
TOTAL NUMBER IN SECOND QUEUE SUBSYSTEM
30
20
10
0
10000
20000
30000
40000
50000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.9



~7,000 observations, batch size = 50;
~140 points total before truncation

Figure 4.10

Table 4.14

Empty Queue Model Output Statistics
10,000 Observation Runs vs. 5000 Observation Runs
CMR Truncation - Consistency Test

MODEL: EMPTY Q

	SAMPLE SIZE: 10,000 OBS (UNBATCHED)			SAMPLE SIZE: 5000 OBS (UNBATCHED)		
RUN	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW
1	0	9.48	0.186	0	11.75	0.299
2	0	8.73	0.139	0	9.27	0.228
3	0	12.90	0.226	0	7.45	0.152
4	0	7.29	0.144	0	6.71	0.204
5	0	10.90	0.169	0	8.03	0.208

Table 4.15

Loaded Queue Model Output Statistics
10,000 Observation Runs vs. 5000 Observation Runs
CMR Truncation - Consistency Test

MODEL: LOADED Q

	SAMPLE SIZE: 10,000 OBS (UNBATCHED)			SAMPLE SIZE: 5000 OBS (UNBATCHED)		
RUN	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW
1	1850	9.93	0.211	1820	14.38	0.393
2	1200	9.41	0.152	1180	11.06	0.278
3	950	14.98	0.247	1260	9.65	0.243
4	1000	10.31	0.224	970	13.40	0.418
5	1600	13.26	0.188	1610	12.62	0.294
9	3950	11.98	0.228	0	54.91	0.936
10	3500	7.15	0.133	0	70.20	1.150

Table 4.16

Queue Series Model Output Statistics
10,000 Observation Runs vs. 5000 Observation Runs
CMR Truncation - Consistency Test

MODEL: Q SERIES

	SAMPLE SIZE: 10,000 OBS (UNBATCHED)			SAMPLE SIZE: 5000 OBS (UNBATCHED)		
RUN	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW
1	0	3.09	0.052	0	3.02	0.075
2	0	3.89	0.064	0	3.91	0.090
3	0	4.13	0.069	0	3.80	0.093
4	0	3.62	0.060	0	3.51	0.081
5	0	3.55	0.063	0	3.50	0.086

Table 4.17

Filling Queue Model Output Statistics
10,000 Observation Runs vs. 5000 Observation Runs
CMR Truncation - Consistency Test

MODEL: FILLING Q

	SAMPLE SIZE: 10,000 OBS (UNBATCHED)			SAMPLE SIZE: 5000 OBS (UNBATCHED)		
RUN	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW
1	100	27.70	0.045	100	27.65	0.061
2	150	27.76	0.044	130	27.48	0.071
3	150	27.70	0.045	150	27.69	0.064
4	150	27.84	0.040	140	27.72	0.060
5	100	27.76	0.041	90	27.67	0.060

Table 4.18

Transient Queue Model Output Statistics
10,000 Observation Runs vs. 5000 Observation Runs
CMR Truncation - Consistency Test

MODEL: TRANSIENT Q

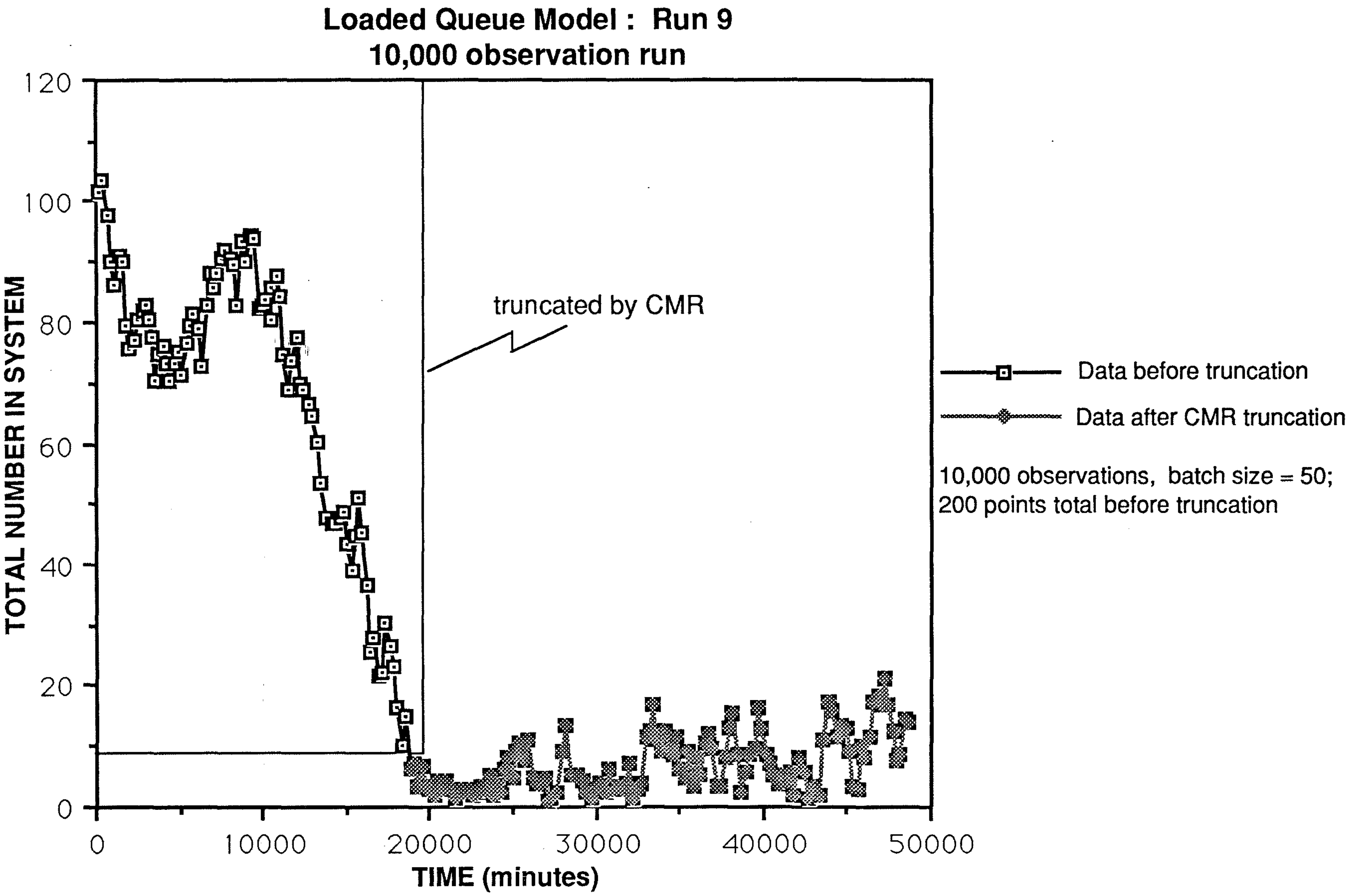
	SAMPLE SIZE: ~7000 OBS * (UNBATCHED)			SAMPLE SIZE: 5000 OBS (UNBATCHED)		
RUN	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW	BATCHES TRUNCATED BY CMR	MEAN	95% CIHW
1	all	n/a	n/a	all	n/a	n/a
2	all	n/a	n/a	all	n/a	n/a
3	all	n/a	n/a	all	n/a	n/a
4	all	n/a	n/a	all	n/a	n/a
5	all	n/a	n/a	all	n/a	n/a

* Unable to reach 10,000 obs.
due to system overflow

mistook for steady state. These data sets are examples in which the consistency of the CMR does not hold. Because of the unexpected behavior of the CMR on their 5000 observation samples, these runs were redone for 10,000 observation samples (along with the others in the consistency test). Figures 4.11 and 4.12 show the graphs of the two new runs. There were two problems with the CMR when it was used on the 5000 observation samples: first, the size of the initial transient period was overwhelmingly large relative to the size of the full data set; second, the behavior of the initial transient period appeared to be relatively stable and settled around a value much higher than the actual steady-state mean. The combined effect was that the CI halfwidth increased as initial data was truncated because the large amount and low variability of the "transient" data counteracted the weight of the transition to actual steady state that occurred near the end of the data set. Therefore, the CI calculations were more strongly (negatively) affected by the reduction in sample size than (positively) by the elimination of initial transients.

On the other hand, when the CMR is used on the same runs with 10,000 observation samples, it correctly determines the optimal truncation point for each sample. It works for these longer runs because the steady-state portion of each data set has become large enough to affect the statistics calculations and, hence, to show that the initial portion is transient. The trickiest part of effectively using the CMR, then, is to determine whether or not the sample being tested is large enough to identify a potentially large initial transient.

One possible way to handle this problem would be to build a check into the CMR that breaks the data set into subsets and analyzes the statistics of the subsets to identify the existence of significant trends. For example, the two Loaded Queue (5000 observation) runs could each be divided into ten sequential pieces, and the mean determined for each piece. A search for trends in the subset statistics would show that the means of the last few subsets are significantly lower than the means of the first subsets. A result of this sort could then trigger a call for more data to be added to the sample so as to clarify the situation. However, this approach is not perfect. If the initial sample size of the two



10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.11

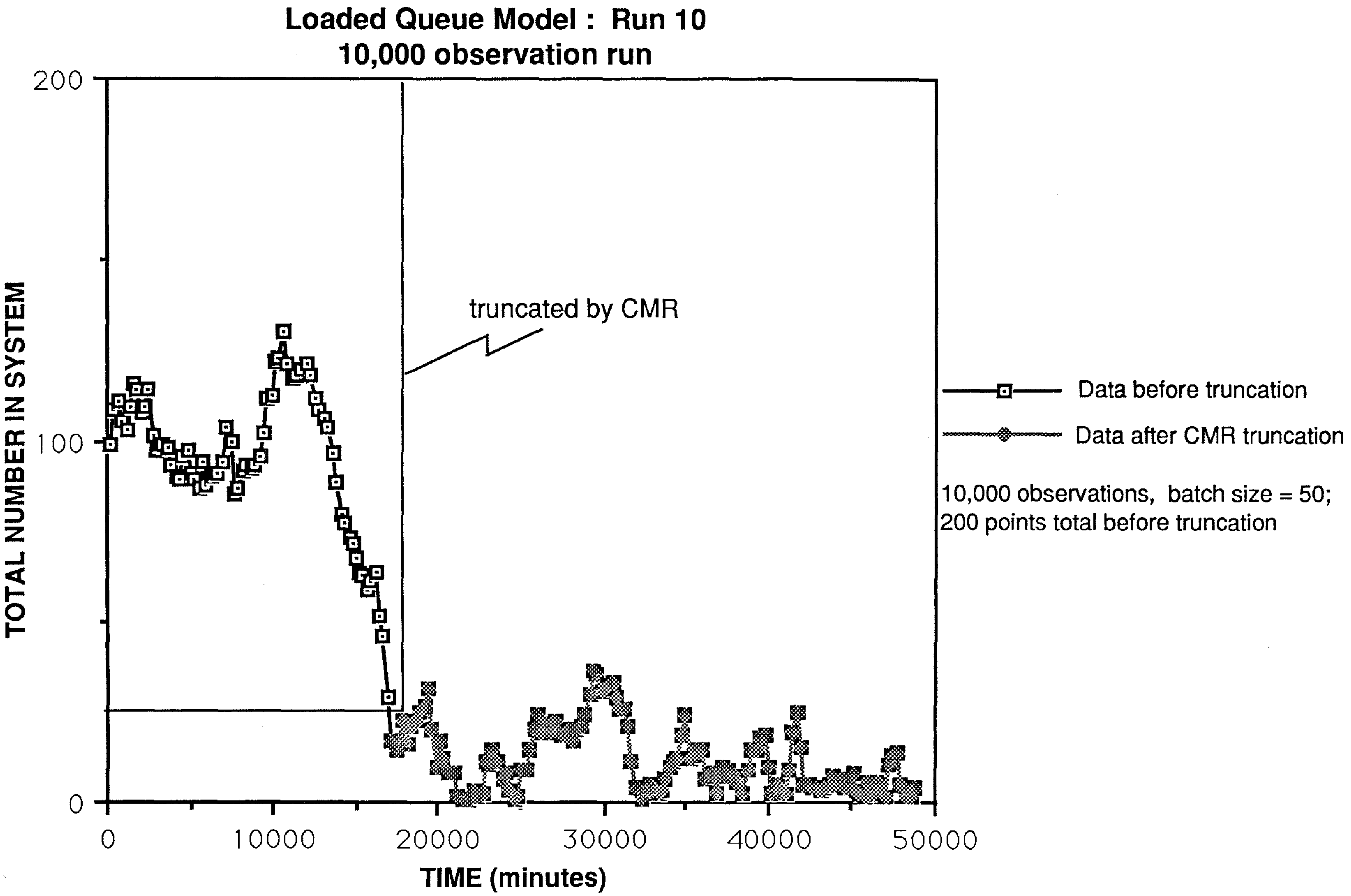

Loaded Queue Model : Run 10
10,000 observation run
200
100
0
TOTAL NUMBER IN SYSTEM
0
10000
20000
30000
40000
50000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
10,000 observations, batch size = 50;
200 points total before truncation

Figure 4.12

Loaded Queue runs was only 2000 observations, the subset statistics would all be consistent; no signal would be given that more data is required. Clearly, the only way to avoid these problems is to collect as much data as possible from the start.

A comparison of the statistics of model run outputs for two different sample sizes, has shown that, in general, the CMR generates fairly consistent results. The optimal truncation point identified for a particular run of 5000 observations is very close to the point identified for the same run of 10,000 observations. For a run with a very long and "stable" transient, the fact that there is an inconsistency between the result for a short run and the result for a long run indicates that there is a problem with the *accuracy* of the short run. In such a case, the only way to generate an acceptable result is to increase the sample size enough to balance the weight of the initial transient in the calculation of statistics.

In general, the CMR tends to be an unusually reliable and consistent methodology for initialization transient detection and truncation. The next section of this chapter will describe how these same confidence maximization concepts are applied to the evaluation of other detection heuristics.

4.3 CONFIDENCE MAXIMIZATION AS AN EVALUATION METHODOLOGY

The initial objective of this project was to find a new, more robust methodology with which to test the effectiveness of initial transient detection and truncation heuristics. This section shows how the Confidence Maximization Procedure is used to compare heuristics' performance; in addition, it outlines tests that were conducted to evaluate the "reliability" of the CMP results.

4.3.1 The Confidence Maximization Procedure

The heuristics chosen for testing were the Ingalls Algorithm and the Crossings-of-the-Mean Rule (see Table 2.1). In addition, the output of extended run data sets, with no truncation, will be checked to study the effectiveness of "diluting" the transients instead of truncating. The two heuristics being tested were chosen for the following reasons. Several of the heuristics listed in Chapter Two require "pre-runs", that is, several test runs to determine the truncation point before it is actually used. These heuristics were rejected for this test because they are not real-time and are unnecessarily wasteful of data; hence, they are not very useful for our purposes of output analysis automation. Aside from the two chosen, the only remaining heuristic is Emshoff and Sisson's Moving-Averages Rule (see Table 2.1), which is essentially a less conservative version of the Ingalls Algorithm. It was decided that testing both Emshoff and Sisson's Rule and the Ingalls Algorithm would be redundant. Because the Ingalls Algorithm is more theoretically sound, it was chosen for testing over Emshoff and Sisson's.

The Ingalls Algorithm is based on cumulative and moving statistics. The cumulative mean of the data set is recomputed at every point, as well as the cumulative standard deviation. A group size and range ratio are specified by the user of the heuristic. The group size chosen for our testing was 30 and the range ratio was 0.25; these values seemed to provide the best overall results. Given the chosen group size, the algorithm looks at the slope of the cumulative mean and standard deviation for each group, moving the group in single point steps for each recalculation. In addition to calculating these slopes for each group, the algorithm determines the maximum and minimum of the cumulative mean and standard deviation values within each group. Next, the algorithm compares the maximum and minimum of the mean and standard deviation to see if these are within the prespecified ratio of each other (minimum * [1.0 + range] > maximum) and determines whether the mean and standard deviation slope values are less than the ratio. If all criteria are met, its definition of "steady state" has been met.

The Crossings-of-the-Mean Rule (hereafter referred to as "Cross/Mean") is a somewhat less elegant procedure, but it is based on common sense. The cumulative mean of the data set is recomputed at every point, and the data value at each point is compared to the mean. Each time the data value changes from greater than to less than the mean or from less than to greater than the mean, one crossing is tallied. Once a pre-specified number of crossings occurs, "steady state" is said to have begun. From some preliminary testing and based on the recommendations in the literature, thirty was chosen as the required number of crossings for our experiment.

The data sets used to illustrate the CMP were the same ten runs of the five models from the first section of this chapter: the Empty Queue, Loaded Queue, Queue Series, Filling Queue, and Transient Queue models. Their 5000 observation samples were used to test the truncation heuristics, while the "no truncation/diluted" samples had 30,000 observations (except for the Transient Queue model, which caused SIMAN to overflow at just over 7000 observations).

The Ingalls and Cross/Mean heuristics were tested by finding the truncation point identified by each for a given data set, and the CMR was then used to determine the "optimal" truncation point and its associated mean and confidence interval. The relative performance of each heuristic is assigned according to which heuristic, on average, yields the smallest CI halfwidth, compared against each other.

Tables 4.19 through 4.23 show the results of all of the truncation heuristics and their associated statistics for each run of each of the five models. The statistics associated with the diluted samples are also listed to show the difference in statistics an increase in sample size can make.

The results of the CMP for the Ingalls Algorithm and the Cross/Mean Rule can be summarized based on these tables. According to the average CI halfwidth for each heuristic with each model, it seems that the Ingalls Algorithm tends to work better than the Cross/Mean Rule with the Loaded Queue and the Filling Queue models; while the Cross/Mean Rule is preferable to the Ingalls Algorithm for the Empty Queue and Queue Series models. (Both work equally well at *not* finding a truncation point for any of the Transient Queue Model runs.) While it may seem that this shows nothing about these two heuristics relative to one another (because they seem to be "even"), that is not exactly true.

Table 4.19
Empty Queue Model Output Statistics
All Four Truncation Approaches
Unbatched Data

MODEL: EMPTY Q UNBATCHED THEOR. MEAN: 9.00

RUN	TRUNC. METHOD	PTS. TRUNC.	SAMPLE SIZE (TOTAL)	PTS. USED FOR STATS	MEAN	HALFWIDTH
[1]	CMR	0	5000	5000	11.75	0.299
	CROSS/MEAN	160	5000	4840	12.01	0.306
	INGALLS	360	5000	4640	12.32	0.315
	EXT. RUN (NO TRUNC.)	0	30000	30000	8.87	0.095
[2]	CMR	0	5000	5000	9.27	0.228
	CROSS/MEAN	160	5000	4840	9.50	0.232
	INGALLS	390	5000	4610	9.86	0.239
	EXT. RUN (NO TRUNC.)	0	30000	30000	8.96	0.100
[3]	CMR	0	5000	5000	7.45	0.152
	CROSS/MEAN	180	5000	4820	7.56	0.156
	INGALLS	390	5000	4610	7.66	0.162
	EXT. RUN (NO TRUNC.)	0	30000	30000	9.36	0.105
[4]	CMR	0	5000	5000	6.71	0.204
	CROSS/MEAN	140	5000	4860	6.81	0.209
	INGALLS	320	5000	4680	6.92	0.216
	EXT. RUN (NO TRUNC.)	0	30000	30000	8.25	0.088
[5]	CMR	0	5000	5000	8.03	0.208
	CROSS/MEAN	90	5000	4910	8.15	0.210
	INGALLS	370	5000	4630	8.15	0.221
	EXT. RUN (NO TRUNC.)	0	30000	30000	10.60	0.111
[6]	CMR	0	5000	5000	6.29	0.149
	CROSS/MEAN	220	5000	4780	6.33	0.154
	INGALLS	380	5000	4620	6.45	0.158
	EXT. RUN (NO TRUNC.)	0	30000	30000	7.70	0.080
[7]	CMR	0	5000	5000	7.25	0.194
	CROSS/MEAN	160	5000	4840	7.39	0.199
	INGALLS	470	5000	4530	7.65	0.210
	EXT. RUN (NO TRUNC.)	0	30000	30000	8.20	0.086
[8]	CMR	0	5000	5000	5.84	0.140
	CROSS/MEAN	50	5000	4950	5.89	0.141
	INGALLS	410	5000	4590	5.95	0.149
	EXT. RUN (NO TRUNC.)	0	30000	30000	8.98	0.099
[9]	CMR	0	5000	5000	20.73	0.591
	CROSS/MEAN	200	5000	4800	21.46	0.607
	INGALLS	430	5000	4570	22.23	0.628
	EXT. RUN (NO TRUNC.)	0	30000	30000	9.65	0.130
[10]	CMR	0	5000	5000	14.67	0.495
	CROSS/MEAN	250	5000	4750	15.11	0.518
	INGALLS	370	5000	4630	15.14	0.531
	EXT. RUN (NO TRUNC.)	0	30000	30000	12.00	0.132

Table 4.20
Loaded Queue Model Output Statistics
All Four Truncation Approaches
Unbatched Data

MODEL: LOADED Q   UNBATCHED   THEOR. MEAN: 9.00

RUN	TRUNC. METHOD	PTS. TRUNC.	SAMPLE SIZE	PTS. USED FOR	MEAN	HALFWIDTH
			(TOTAL)	STATS		
[1]	CMR	1820	5000	3180	14.38	0.393
	CROSS/MEAN	260	5000	4740	32.03	0.818
	INGALLS	150	5000	4850	33.49	0.843
	EXT. RUN (NO TRUNC.)	0	30000	30000	12.80	0.202
[2]	CMR	1180	5000	3820	11.06	0.278
	CROSS/MEAN	110	5000	4890	22.01	0.652
	INGALLS	400	5000	4600	18.53	0.551
	EXT. RUN (NO TRUNC.)	0	30000	30000	11.50	0.163
[3]	CMR	1260	5000	3740	9.65	0.243
	CROSS/MEAN	all	5000	0	---	---
	INGALLS	150	5000	4850	20.03	0.642
	EXT. RUN (NO TRUNC.)	0	30000	30000	11.80	0.167
[4]	CMR	970	5000	4030	13.44	0.418
	CROSS/MEAN	440	5000	4560	20.10	0.682
	INGALLS	80	5000	4920	25.47	0.829
	EXT. RUN (NO TRUNC.)	0	30000	30000	11.60	0.180
[5]	CMR	1610	5000	3390	12.62	0.294
	CROSS/MEAN	210	5000	4790	28.04	0.775
	INGALLS	150	5000	4850	29.03	0.804
	EXT. RUN (NO TRUNC.)	0	30000	30000	14.60	0.197
[6]	CMR	1230	5000	3770	6.38	0.158
	CROSS/MEAN	all	5000	0	---	---
	INGALLS	4490	5000	510	6.04	0.434
	EXT. RUN (NO TRUNC.)	0	30000	30000	9.63	0.131
[7]	CMR	2960	5000	2040	5.02	0.175
	CROSS/MEAN	110	5000	4890	38.27	0.987
	INGALLS	260	5000	4740	36.39	0.971
	EXT. RUN (NO TRUNC.)	0	30000	30000	13.70	0.227
[8]	CMR	2510	5000	2490	6.83	0.197
	CROSS/MEAN	550	5000	4450	29.19	0.867
	INGALLS	220	5000	4780	34.82	0.998
	EXT. RUN (NO TRUNC.)	0	30000	30000	14.60	0.234
[9]	CMR	0	5000	5000	54.91	0.936
	CROSS/MEAN	1230	5000	3770	46.24	1.094
	INGALLS	250	5000	4750	52.75	0.945
	EXT. RUN (NO TRUNC.)	0	30000	30000	15.30	0.262
[10]	CMR	0	5000	5000	70.20	1.150
	CROSS/MEAN	260	5000	4740	68.30	1.190
	INGALLS	160	5000	4840	69.00	1.180
	EXT. RUN (NO TRUNC.)	0	30000	30000	21.20	0.330

Table 4.21
Queue Series Model Output Statistics
All Four Truncation Approaches
Unbatched Data

MODEL: SERIES Q UNBATCHED THEOR. MEAN: 5.3608

RUN	TRUNC. METHOD	PTS. TRUNC.	SAMPLE SIZE (TOTAL)	PTS. USED FOR STATS	MEAN	HALFWIDTH
[1]	CMR	0	5000	5000	3.02	0.075
	CROSS/MEAN	100	5000	4900	3.05	0.076
	INGALLS	370	5000	4630	3.12	0.079
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.34	0.036
[2]	CMR	0	5000	5000	3.91	0.090
	CROSS/MEAN	40	5000	4600	3.94	0.090
	INGALLS	360	5000	4640	4.00	0.094
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.83	0.037
[3]	CMR	0	5000	5000	3.80	0.093
	CROSS/MEAN	150	5000	4850	3.87	0.095
	INGALLS	270	5000	4730	3.90	0.097
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.89	0.038
[4]	CMR	0	5000	5000	3.51	0.081
	CROSS/MEAN	60	5000	4940	3.54	0.082
	INGALLS	330	5000	4670	3.60	0.085
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.79	0.036
[5]	CMR	0	5000	5000	3.50	0.086
	CROSS/MEAN	50	5000	4950	3.52	0.086
	INGALLS	300	5000	4700	3.60	0.090
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.47	0.035
[6]	CMR	0	5000	5000	3.46	0.078
	CROSS/MEAN	210	5000	4790	3.54	0.081
	INGALLS	420	5000	4580	3.57	0.083
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.70	0.036
[7]	CMR	0	5000	5000	3.72	0.091
	CROSS/MEAN	90	5000	4910	3.75	0.093
	INGALLS	310	5000	4690	3.81	0.096
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.53	0.034
[8]	CMR	0	5000	5000	3.81	0.092
	CROSS/MEAN	140	5000	4860	3.86	0.094
	INGALLS	390	5000	4610	3.93	0.097
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.72	0.036
[9]	CMR	0	5000	5000	3.45	0.086
	CROSS/MEAN	60	5000	4940	3.48	0.087
	INGALLS	360	5000	4640	3.55	0.091
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.62	0.036
[10]	CMR	0	5000	5000	3.38	0.084
	CROSS/MEAN	70	5000	4930	3.40	0.085
	INGALLS	280	5000	4720	3.49	0.088
	EXT. RUN (NO TRUNC.)	0	30000	30000	3.44	0.034

Table 4.22
Filling Queue Model Output Statistics
All Four Truncation Approaches
Unbatched Data

MODEL: FILLING Q　　UNBATCHED　　THEOR. MEAN: unknown

RUN	TRUNC. METHOD	PTS. TRUNC.	SAMPLE SIZE (TOTAL)	PTS. USED FOR STATS	MEAN	HALFWIDTH
[1]	CMR	100	5000	4900	27.65	0.061
	CROSS/MEAN	780	5000	4220	27.61	0.067
	INGALLS	410	5000	4590	27.67	0.063
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.028
[2]	CMR	130	5000	4870	27.48	0.071
	CROSS/MEAN	700	5000	4300	27.43	0.078
	INGALLS	650	5000	4350	27.40	0.078
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.030
[3]	CMR	150	5000	4850	27.69	0.064
	CROSS/MEAN	1160	5000	3840	27.65	0.073
	INGALLS	860	5000	4140	27.58	0.072
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.60	0.029
[4]	CMR	140	5000	4860	27.72	0.060
	CROSS/MEAN	830	5000	4170	27.75	0.065
	INGALLS	660	5000	4340	27.70	0.065
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.029
[5]	CMR	90	5000	4910	27.67	0.060
	CROSS/MEAN	500	5000	4500	27.77	0.060
	INGALLS	410	5000	4590	27.76	0.060
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.60	0.028
[6]	CMR	100	5000	4900	27.33	0.071
	CROSS/MEAN	710	5000	4290	27.22	0.079
	INGALLS	310	5000	4690	27.30	0.074
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.029
[7]	CMR	110	5000	4890	27.67	0.058
	CROSS/MEAN	570	5000	4430	27.62	0.061
	INGALLS	270	5000	4730	27.65	0.059
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.030
[8]	CMR	110	5000	4890	27.11	0.081
	CROSS/MEAN	800	5000	4200	27.01	0.091
	INGALLS	330	5000	4670	27.09	0.084
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.20	0.033
[9]	CMR	210	5000	4790	27.73	0.061
	CROSS/MEAN	870	5000	4130	27.68	0.068
	INGALLS	370	5000	4630	27.72	0.062
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.40	0.030
[10]	CMR	130	5000	4870	27.30	0.072
	CROSS/MEAN	740	5000	4260	27.24	0.079
	INGALLS	630	5000	4370	27.24	0.078
	EXT. RUN (NO TRUNC.)	0	30000	30000	27.50	0.028

Table 4.23
Transient Queue Model Output Statistics
All Four Truncation Approaches
Unbatched Data

MODEL: TRANSIENT Q UNBATCHED THEOR. MEAN: none

RUN	TRUNC. METHOD	PTS. TRUNC.	SAMPLE SIZE (TOTAL)	PTS. USED FOR STATS	MEAN	HALFWIDTH
[1]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7050	30000	1298	17.5
[2]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7204	30000	1355	17.1
[3]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7040	30000	1286	17.3
[4]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7114	30000	1286	17.7
[5]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	6702	30000	1292	17.9
[6]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7252	30000	1363	17.6
[7]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	6942	30000	1278	17.6
[8]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	6926	30000	1303	17.8
[9]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7212	30000	1296	16.9
[10]	CMR	all	5000	0	--	--
	CROSS/MEAN	all	5000	0	--	--
	INGALLS	all	5000	0	--	--
	EXT. RUN (NO TRUNC.)	0	7012	30000	1261	18

Rather, this test shows that each heuristic works better than the other for a particular class of data. Therefore, the tentative conclusion to be drawn from the CMP is that the Ingalls Algorithm is more effective for data that has a stronger initial transient skew (as the Loaded Queue and Filling Queue models have), and the Cross/Mean Rule works better for data with no significant initial transient (as the Empty Queue and Queue Series models have).

4.3.2 Reliability of the Confidence Maximization Procedure

Given the above "conclusion", in order to evaluate how well the CMP has determined the "best" heuristic (assuming that a different "best" heuristic can be associated with each class of output), the reliability of the results will be assessed by looking, once again, at coverages. The reliability of the CMP can be considered good if the coverages associated with the "best" heuristic (for each class of output) are as good or better than the coverages for the other heuristics. Coverages for each heuristic were calculated in the same manner as the previous section in this chapter, and Tables 4.24 through 4.28 show the results of these calculations.

These results show that the coverages with the Ingalls truncation points are better than with the Cross/Mean truncation points for the Loaded Queue and Filling Queue models (although for the runs tested here, coverage values generally are not very high due to the sample size constraints). This result is in agreement with the CMP's conclusion that Ingalls works better than Cross/Mean for data sets with an initial transient. For the Empty Queue and Queue Series models, the coverage using the Cross/Mean truncation points is about the same as the coverage using the Ingalls truncation points. Since coverage is unaffected while CI halfwidth is improved by using Cross/Mean rather than Ingalls, the CMP's conclusion that Cross/Mean works better than Ingalls for data without an initial transient is confirmed by this test. Therefore, based on coverage comparisons, the results of the CMP are reliable.

Table 4.24
Empty Queue Model Coverage - Batched Data - All Truncation Approaches

MODEL: EMPTY Q ACTUAL MEAN: 9.00 100 OBS/ BATCH

RUN	DET. METHOD	BATCHES TRUNC.	TOTAL SAMPLE SIZE (BATCHES)	% TRUNCATED (FROM TOTAL)	MEAN	95% C.I. H.W.	COVERAGE
[1]	CMR	0	50	0.00	11.80	2.93	+
	CROSS/MEAN	2	50	0.04	12.10	3.10	+
	INGALLS	4	50	0.08	12.40	3.20	-
	EXT. RUN	0	300	0.00	8.87	0.87	+
[2]	CMR	0	50	0.00	9.28	2.10	+
	CROSS/MEAN	2	50	0.04	9.57	2.21	+
	INGALLS	4	50	0.08	9.87	2.26	+
	EXT. RUN	0	300	0.00	8.96	0.93	+
[3]	CMR	0	50	0.00	7.45	1.24	-
	CROSS/MEAN	2	50	0.04	7.57	1.31	-
	INGALLS	4	50	0.08	7.67	1.36	+
	EXT. RUN	0	300	0.00	9.36	0.99	+
[4]	CMR	0	50	0.00	6.71	1.97	-
	CROSS/MEAN	1	50	0.02	6.79	2.00	-
	INGALLS	3	50	0.06	6.92	2.03	-
	EXT. RUN	0	300	0.00	8.25	0.81	+
[5]	CMR	0	50	0.00	8.03	1.96	+
	CROSS/MEAN	1	50	0.02	8.17	2.03	+
	INGALLS	4	50	0.08	8.16	2.18	+
	EXT. RUN	0	300	0.00	10.60	1.05	-
[6]	CMR	0	50	0.00	6.29	1.21	-
	CROSS/MEAN	2	50	0.04	6.34	1.28	-
	INGALLS	4	50	0.08	6.46	1.33	-
	EXT. RUN	0	300	0.00	7.70	0.72	-
[7]	CMR	0	50	0.00	7.25	1.79	+
	CROSS/MEAN	2	50	0.04	7.42	1.90	+
	INGALLS	5	50	0.10	7.69	2.01	+
	EXT. RUN	0	300	0.00	8.20	0.78	-
[8]	CMR	0	50	0.00	5.84	1.12	-
	CROSS/MEAN	1	50	0.02	5.93	1.16	-
	INGALLS	4	50	0.08	5.96	1.22	-
	EXT. RUN	0	300	0.00	8.98	0.92	+
[9]	CMR	0	50	0.00	20.70	6.03	-
	CROSS/MEAN	2	50	0.04	21.50	6.36	-
	INGALLS	4	50	0.08	22.10	6.58	-
	EXT. RUN	0	300	0.00	9.65	1.26	+
[10]	CMR	0	50	0.00	14.70	5.02	-
	CROSS/MEAN	3	50	0.06	15.20	5.46	-
	INGALLS	4	50	0.08	15.20	5.58	-
	EXT. RUN	0	300	0.00	12.00	1.26	-

	CMR	CROSS/MEAN	INGALLS	EXT. RUN
AVG. COVERAGE:	40%	40%	40%	60.00%

Table 4.25
Loaded Queue Model Coverage - Batched Data - All Truncation Approaches

MODEL: LOADEDQ ACTUAL MEAN: 9.00 100 OBS/ BATCH

RUN	DET. METHOD	BATCHES TRUNC.	TOTAL SAMPLE SIZE (BATCHES)	% TRUNCATED (FROM TOTAL)	MEAN	95% C.I. H.W.	COVERAGE
[1]	CMR	18	50	0.36	14.50	4.09	-
	CROSS/MEAN	3	50	0.06	31.40	8.46	-
	INGALLS	2	50	0.04	32.80	8.78	-
	EXT. RUN	0	300	0.00	12.80	1.99	-
[2]	CMR	12	50	0.24	11.00	2.66	+
	CROSS/MEAN	1	50	0.02	22.20	6.87	-
	INGALLS	4	50	0.08	18.50	5.75	-
	EXT. RUN	0	300	0.00	11.50	1.59	-
[3]	CMR	13	50	0.26	9.61	2.29	+
	CROSS/MEAN	all	50	1.00	--	--	n/a
	INGALLS	2	50	0.04	19.30	6.41	-
	EXT. RUN	0	300	0.00	11.80	1.36	-
[4]	CMR	10	50	0.20	13.40	4.40	+
	CROSS/MEAN	4	50	0.08	20.70	7.38	-
	INGALLS	1	50	0.02	25.20	8.66	-
	EXT. RUN	0	300	0.00	11.60	1.77	-
[5]	CMR	16	50	0.32	12.60	3.00	-
	CROSS/MEAN	2	50	0.04	28.20	8.24	-
	INGALLS	2	50	0.04	28.20	8.24	-
	EXT. RUN	0	300	0.00	14.60	1.95	-
[6]	CMR	12	50	0.24	6.51	1.36	-
	CROSS/MEAN	all	50	1.00	--	--	n/a
	INGALLS	45	50	0.90	5.88	12.70	+
	EXT. RUN	0	300	0.00	9.63	1.26	+
[7]	CMR	30	50	0.60	4.97	1.44	-
	CROSS/MEAN	1	50	0.02	38.40	10.50	-
	INGALLS	3	50	0.06	35.90	10.20	-
	EXT. RUN	0	300	0.00	13.70	2.25	-
[8]	CMR	25	50	0.50	6.86	1.59	-
	CROSS/MEAN	6	50	0.12	28.30	8.92	-
	INGALLS	2	50	0.04	35.20	10.70	-
	EXT. RUN	0	300	0.00	14.60	2.32	-
[9]	CMR	0	50	0.00	54.90	9.63	-
	CROSS/MEAN	12	50	0.24	46.50	11.60	-
	INGALLS	2	50	0.04	53.10	9.94	-
	EXT. RUN	0	300	0.00	15.30	2.61	-
[10]	CMR	0	50	0.00	70.20	11.90	-
	CROSS/MEAN	3	50	0.06	67.90	12.70	-
	INGALLS	2	50	0.04	68.70	12.50	-
	EXT. RUN	0	300	0.00	21.20	3.29	-

	CMR	CROSS/MEAN	INGALLS	EXT. RUN
AVG. COVERAGE:	30%	0%	10%	10.00%

Table 4.26
Queue Series Model Coverage - Batched Data - All Truncation Approaches

MODEL: SERIES Q THEOR. MEAN: 5.36 100 OBS/ BATCH

RUN	DET. METHOD	BATCHES TRUNC.	TOTAL SAMPLE SIZE (BATCHES)	% TRUNCATED (FROM TOTAL)	MEAN	95% C.I. H.W.	MEAN OF EXT. RUN MEANS	COVERAGE (EX. RUN MN. & HW)
[1]	CMR	0	50	0.00	3.02	0.361	3.633	- (0%)
	CROSS/MEAN	1	50	0.02	3.05	0.361	3.633	- (0%)
	INGALLS	4	50	0.08	3.13	0.370	3.633	- (0%)
	EXT. RUN	0	300	0.00	3.34	0.207	3.633	- (40%)
[2]	CMR	0	50	0.00	3.91	0.554	3.633	+
	CROSS/MEAN	0	50	0.00	3.91	0.554	3.633	+
	INGALLS	4	50	0.08	4.02	0.581	3.633	+
	EXT. RUN	0	300	0.00	3.83	0.227	3.633	- (35%)
[3]	CMR	0	50	0.00	3.80	0.657	3.633	+
	CROSS/MEAN	2	50	0.04	3.90	0.665	3.633	+
	INGALLS	3	50	0.06	3.88	0.678	3.633	+
	EXT. RUN	0	300	0.00	3.89	0.245	3.633	- (42%)
[4]	CMR	0	50	0.00	3.51	0.460	3.633	+
	CROSS/MEAN	1	50	0.02	3.56	0.448	3.633	+
	INGALLS	3	50	0.06	3.60	0.474	3.633	+
	EXT. RUN	0	300	0.00	3.79	0.211	3.633	- (74%)
[5]	CMR	0	50	0.00	3.50	0.642	3.633	+
	CROSS/MEAN	1	50	0.02	3.54	0.650	3.633	+
	INGALLS	3	50	0.06	3.60	0.672	3.633	+
	EXT. RUN	0	300	0.00	3.47	0.222	3.633	- (76%)
[6]	CMR	0	50	0.00	3.46	0.511	3.633	+
	CROSS/MEAN	2	50	0.04	3.54	0.521	3.633	+
	INGALLS	4	50	0.08	3.58	0.540	3.633	+
	EXT. RUN	0	300	0.00	3.70	0.231	3.633	+
[7]	CMR	0	50	0.00	3.72	0.680	3.633	+
	CROSS/MEAN	1	50	0.02	3.75	0.690	3.633	+
	INGALLS	3	50	0.06	3.81	0.716	3.633	+
	EXT. RUN	0	300	0.00	3.53	0.213	3.633	+
[8]	CMR	0	50	0.00	3.81	0.620	3.633	+
	CROSS/MEAN	1	50	0.02	3.85	0.626	3.633	+
	INGALLS	4	50	0.08	3.93	0.657	3.633	+
	EXT. RUN	0	300	0.00	3.72	0.222	3.633	+
[9]	CMR	0	50	0.00	3.45	0.621	3.633	+
	CROSS/MEAN	1	50	0.02	3.49	0.631	3.633	+
	INGALLS	4	50	0.08	3.57	0.666	3.633	+
	EXT. RUN	0	300	0.00	3.62	0.225	3.633	+
[10]	CMR	0	50	0.00	3.38	0.519	3.633	+
	CROSS/MEAN	1	50	0.02	3.42	0.523	3.633	+
	INGALLS	3	50	0.06	3.48	0.538	3.633	+
	EXT. RUN	0	300	0.00	3.44	0.210	3.633	- (58%)

	CMR	CROSS/MEAN	INGALLS	EXT. RUN	EXT. RUN CIHW [3.520,3.746]
AVG. COVERAGE:	90%	90%	90%	72.50%	

Table 4.27
Filling Queue Model Coverage - Batched Data - All Truncation Approaches

MODEL: FILLING Q ACTUAL MEAN: unknown 100 OBS/ BATCH

RUN	DET. METHOD	BATCHES TRUNC.	TOTAL SAMPLE SIZE (BATCHES)	% TRUNC. (FROM TOT.)	MEAN	95% C.I. H.W.	MEAN OF EXT. RUN MEANS	COVERAGE (EX. RUN MN. & HW)
[1]	CMR	1	50	0.02	27.60	0.291	27.5	+
	CROSS/MEAN	8	50	0.16	27.60	0.335	27.5	+
	INGALLS	4	50	0.08	27.70	0.311	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.156	27.5	+
[2]	CMR	1	50	0.02	27.40	0.425	27.5	+
	CROSS/MEAN	7	50	0.14	27.40	0.476	27.5	+
	INGALLS	7	50	0.14	27.40	0.476	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.188	27.5	+
[3]	CMR	2	50	0.04	27.70	0.351	27.5	+
	CROSS/MEAN	12	50	0.24	27.60	0.436	27.5	+
	INGALLS	9	50	0.18	27.60	0.406	27.5	+
	EXT. RUN	0	300	0.00	27.60	0.180	27.5	+
[4]	CMR	1	50	0.02	27.70	0.299	27.5	+
	CROSS/MEAN	8	50	0.16	27.80	0.303	27.5	- (38%)
	INGALLS	7	50	0.14	27.70	0.301	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.178	27.5	+
[5]	CMR	1	50	0.02	27.70	0.338	27.5	+
	CROSS/MEAN	5	50	0.10	27.80	0.319	27.5	- (49%)
	INGALLS	4	50	0.08	27.70	0.315	27.5	+
	EXT. RUN	0	300	0.00	27.60	0.162	27.5	- (80%)
[6]	CMR	1	50	0.02	27.30	0.420	27.5	+
	CROSS/MEAN	7	50	0.14	27.20	0.430	27.5	+
	INGALLS	3	50	0.06	27.30	0.447	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.176	27.5	+
[7]	CMR	1	50	0.02	27.70	0.309	27.5	+
	CROSS/MEAN	6	50	0.12	27.60	0.347	27.5	+
	INGALLS	3	50	0.06	27.70	0.341	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.187	27.5	+
[8]	CMR	1	50	0.02	27.10	0.597	27.5	+
	CROSS/MEAN	8	50	0.16	27.00	0.703	27.5	+
	INGALLS	3	50	0.06	27.10	0.632	27.5	+
	EXT. RUN	0	300	0.00	27.20	0.221	27.5	- (8%)
[9]	CMR	2	50	0.04	27.70	0.350	27.5	+
	CROSS/MEAN	9	50	0.18	27.70	0.419	27.5	+
	INGALLS	4	50	0.08	27.70	0.374	27.5	+
	EXT. RUN	0	300	0.00	27.40	0.188	27.5	+
[10]	CMR	1	50	0.02	27.30	0.383	27.5	+
	CROSS/MEAN	7	50	0.14	27.20	0.437	27.5	+
	INGALLS	6	50	0.12	27.30	0.427	27.5	+
	EXT. RUN	0	300	0.00	27.50	0.158	27.5	+

	CMR	CROSS/MEAN	INGALLS	EXT. RUN	EXT. RUN CIHW [27.41,27.55]
AVG. COVERAGE:	100%	88.70%	100%	88.80%	

Table 4.28
Transient Queue Model Coverage - Batched - All Truncation Approaches

MODEL: TRANS Q ACTUAL MEAN: NONE 100 OBS/ BATCH

RUN	DET. METHOD	BATCHES TRUNC.	TOTAL SAMPLE SIZE (BATCHES)	% TRUNC. (FROM TOT.)	MEAN	95% C.I. H.W.	MEAN OF EXT. RUN MEANS	COVERAGE (EX. RUN MN. & HW)
[1]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	70	0	1289	178	1287.2	-
[2]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	72	0	1354	175	1287.2	-
[3]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	70	0	1279	177	1287.2	-
[4]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	71	0	1284	181	1287.2	-
[5]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	67	0	1291	184	1287.2	-
[6]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	72	0	1354	180	1287.2	-
[7]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	69	0	1270	180	1287.2	-
[8]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	69	0	1298	182	1287.2	-
[9]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	72	0	1294	173	1287.2	-
[10]	CMR	all	50	100	n/a	n/a	1287.2	n/a
	CROSS/MEAN	all	50	100	n/a	n/a	1287.2	n/a
	INGALLS	all	50	100	n/a	n/a	1287.2	n/a
	EXT. RUN	0	70	0	1259	184	1287.2	-

	CMR	CROSS/MEAN	INGALLS	EXT. RUN	EXT. RUN CIHW [1277.3,1317.1]
AVG. COVERAGE:	n/a	n/a	n/a	0.00%	

4.4 THE EFFECTIVENESS OF THE CONFIDENCE MAXIMIZATION RULE

Unfortunately, using the CMP to compare only the Cross/Mean and Ingalls heuristics would not be very helpful to the average simulationist because, although the results generated are perfectly valid in a theoretical sense, the "answer" is different depending on the form of the output data. This is a problem because the user does not always have prescience regarding the form of his output; he may not know whether or not to expect an initial transient in his output data. After all, this is probably why he wants to use a transient detection heuristic in the first place! In addition to this problem, there is also the issue that, both in general and compared with the results of the CMR, neither heuristic performs very well in terms of CI halfwidth and coverage for any of the steady-state models (i.e., all but the Transient Queue Model). The CMR outperforms both heuristics for all four steady-state models; its average CI halfwidths are smaller and its coverage values are at least as good or better in all cases.

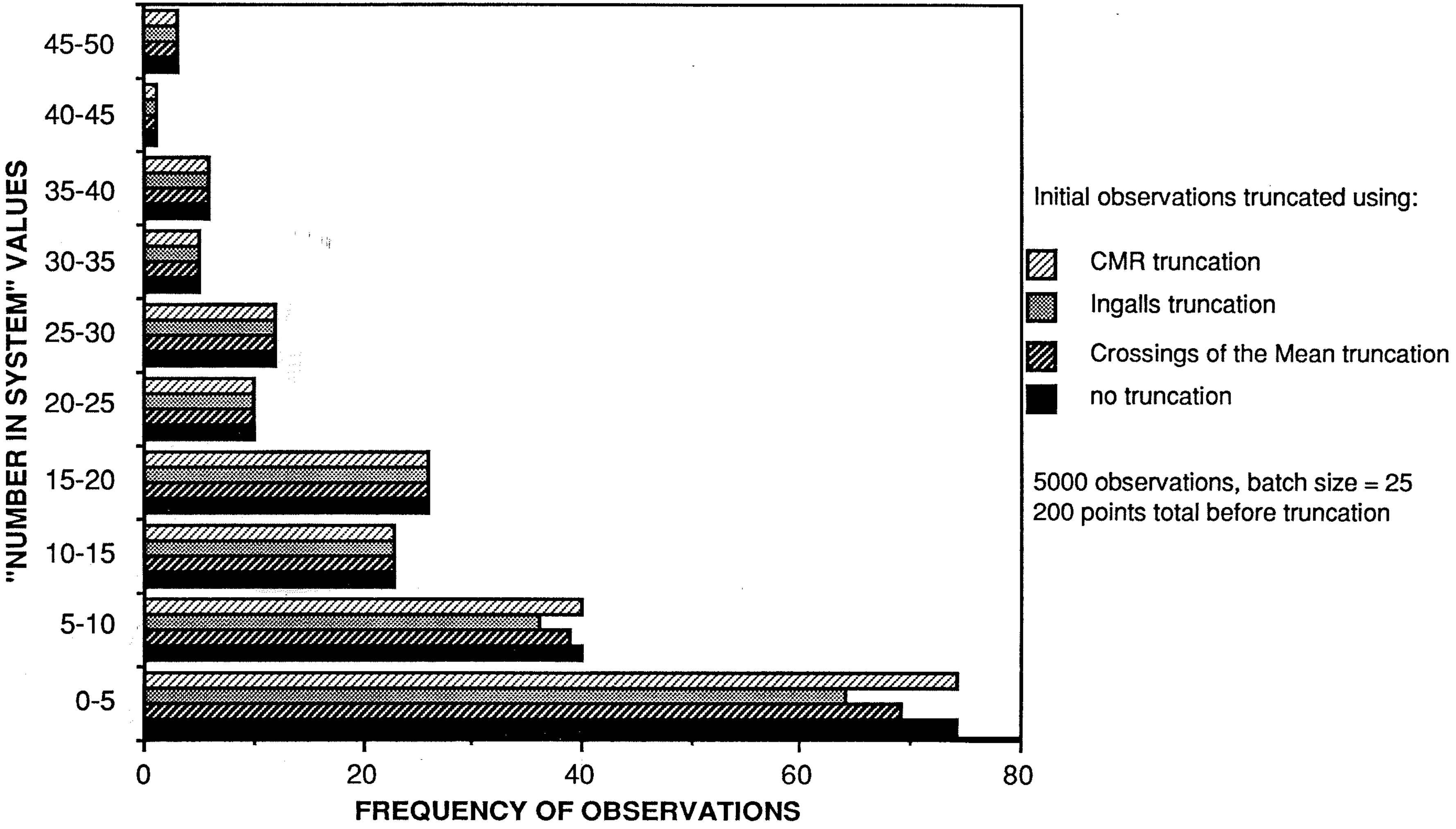
The only method for transient elimination that outperforms the CMR here is to make the sample sizes six times larger (30,000 observations instead of 5000) and let the transients be diluted by the large steady-state periods. It should be noted that, even with a run as long as 30,000 observations, the Loaded Queue Model statistics still have a lower coverage value than they do with CMR truncation. This is true because the transient is much stronger for this model than for any of the others and, therefore, requires an even longer run for the dilution to be effective. Although the dilution approach seems to generate reliable statistics, its use will be discounted here on the basis of impracticality. In many cases, the need for such a large sample size in order for dilution to work makes the use of some sort of truncation method a preferable approach.

Hence, it may be concluded that of the three transient detection/truncation heuristics tested here, the CMR is the best approach. Its statistical superiority over Cross/Mean and Ingalls may be partially explained by the fact that the two latter heuristics use the cumulative mean of the system variable as the basis for steady-state detection. Because of the high variability common in simulation output, the cumulative mean takes a while to settle, even if the initial data is not heavily biased. The extra time required to allow for settling of the cumulative mean causes heuristics like Cross/Mean and Ingalls to generate overly conservative estimates of the onset of steady state.

The effects of overconservative estimation can perhaps be seen more clearly in the frequency bargraphs displayed in Figures 4.13 through 4.17. Bargraphs have been developed for one run of each of the five models. The graphs show the distribution of output data by separating the "number in system" observations according to their magnitude and computing the frequency of observations in each "value region" (the abscissa). Note that the "value regions" are defined differently for each model because the range of values associated with each model's output varies widely. For example, the highest observation of "number in system" for this run of the Loaded Queue Model is nearly 120, while for the run of the Queue Series Model, the highest value is only twelve. To moderate the range requirements for these graphs, the raw data (5000 observations, once again) was batched by groups of 25 to create the data sets used in the frequency counts. This batching does not affect the general shapes of the distributions nor does it alter the effects of truncation on these distributions. In these plots, each truncation heuristic is assigned a bar within each value region, so each value region has four bars: one for no truncation, to illustrate the "raw" distribution of the data; one for the data distribution after CMR truncation has been applied to the "raw" data; one for the distribution after Cross/Mean truncation has been applied; and one for the distribution after Ingalls truncation has been applied.

The purpose of these plots is to illustrate the "smoothing" effects of truncation on the output data distributions. Clearly, because the optimal CMR truncation point for the Empty Queue and Queue Series models is zero, for these models, the CMR distribution bars are the same as those for the no truncation case. Note also that for the Transient Queue Model's output, none of the heuristics identified a truncation point. Its frequency

Empty Queue Model: Run 1
Frequency of Batched Observations Within
Each Value Range



5000 observations, batch size = 25
200 points total before truncation

Figure 4.13

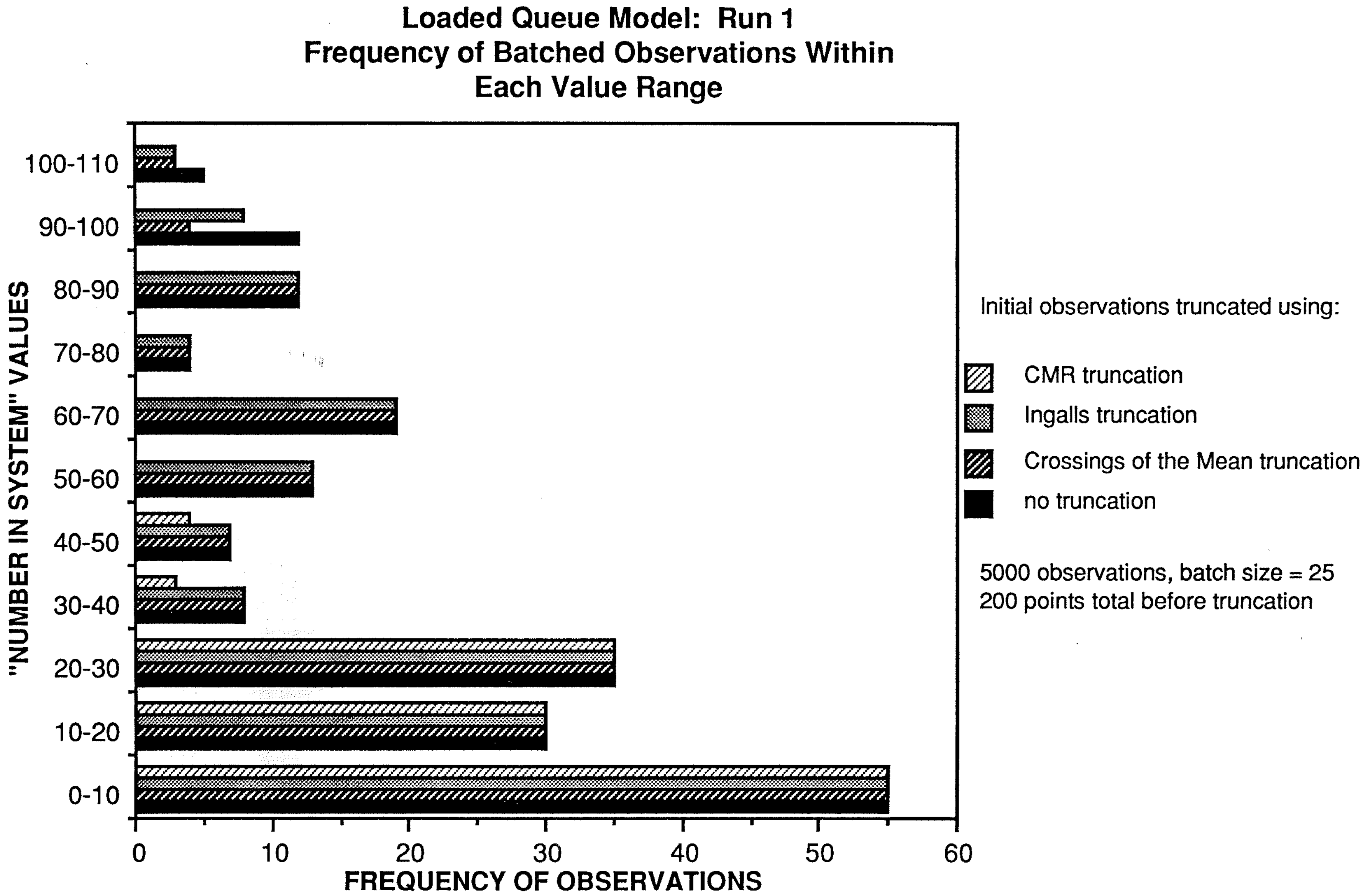
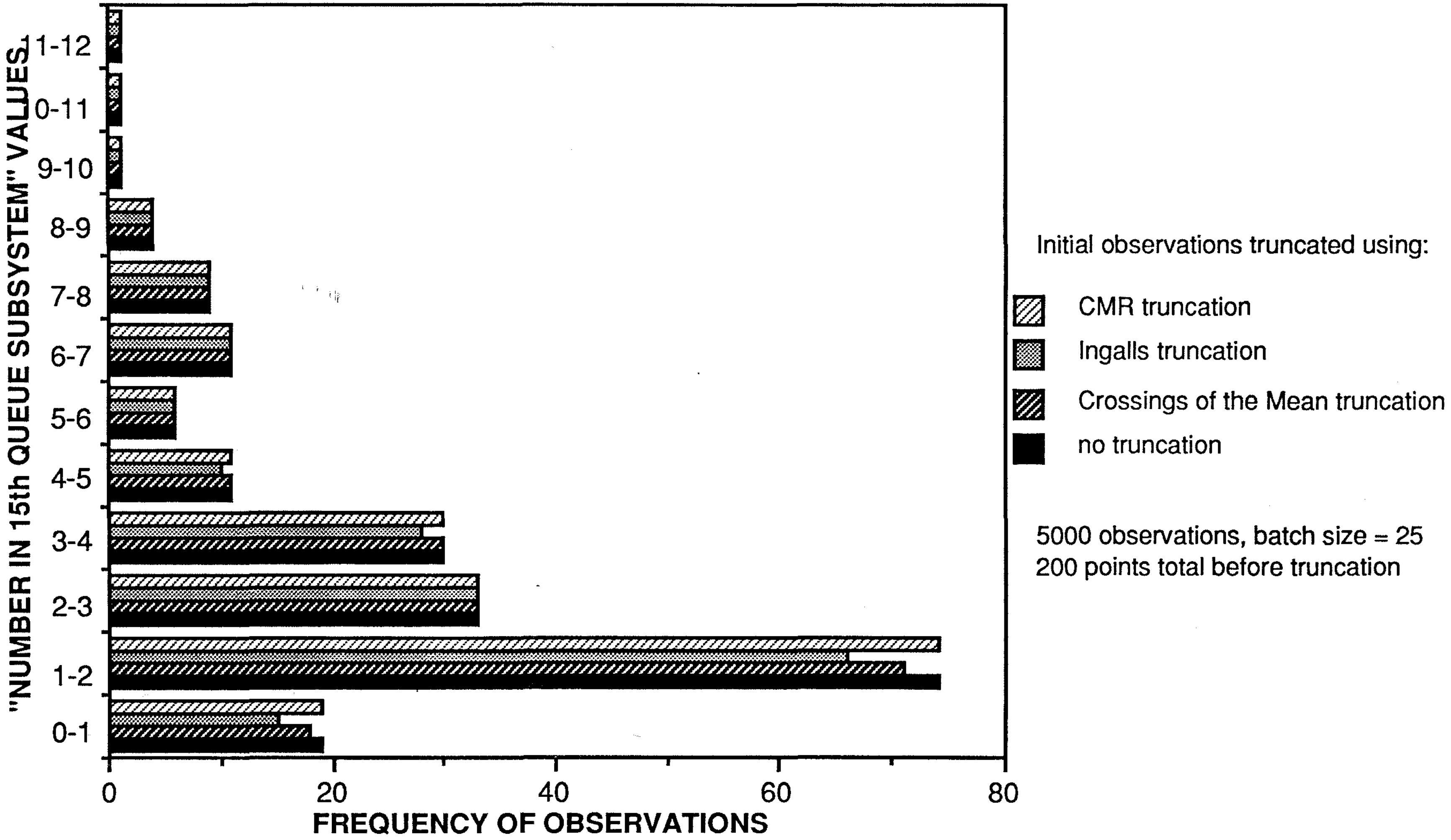

Loaded Queue Model: Run 1
Frequency of Batched Observations Within
Each Value Range
"NUMBER IN SYSTEM" VALUES
100-110
90-100
80-90
70-80
60-70
50-60
40-50
30-40
20-30
10-20
0-10
0
10
20
30
40
50
60
FREQUENCY OF OBSERVATIONS
Initial observations truncated using:
CMR truncation
Ingalls truncation
Crossings of the Mean truncation
no truncation
5000 observations, batch size = 25
200 points total before truncation

Figure 4.14

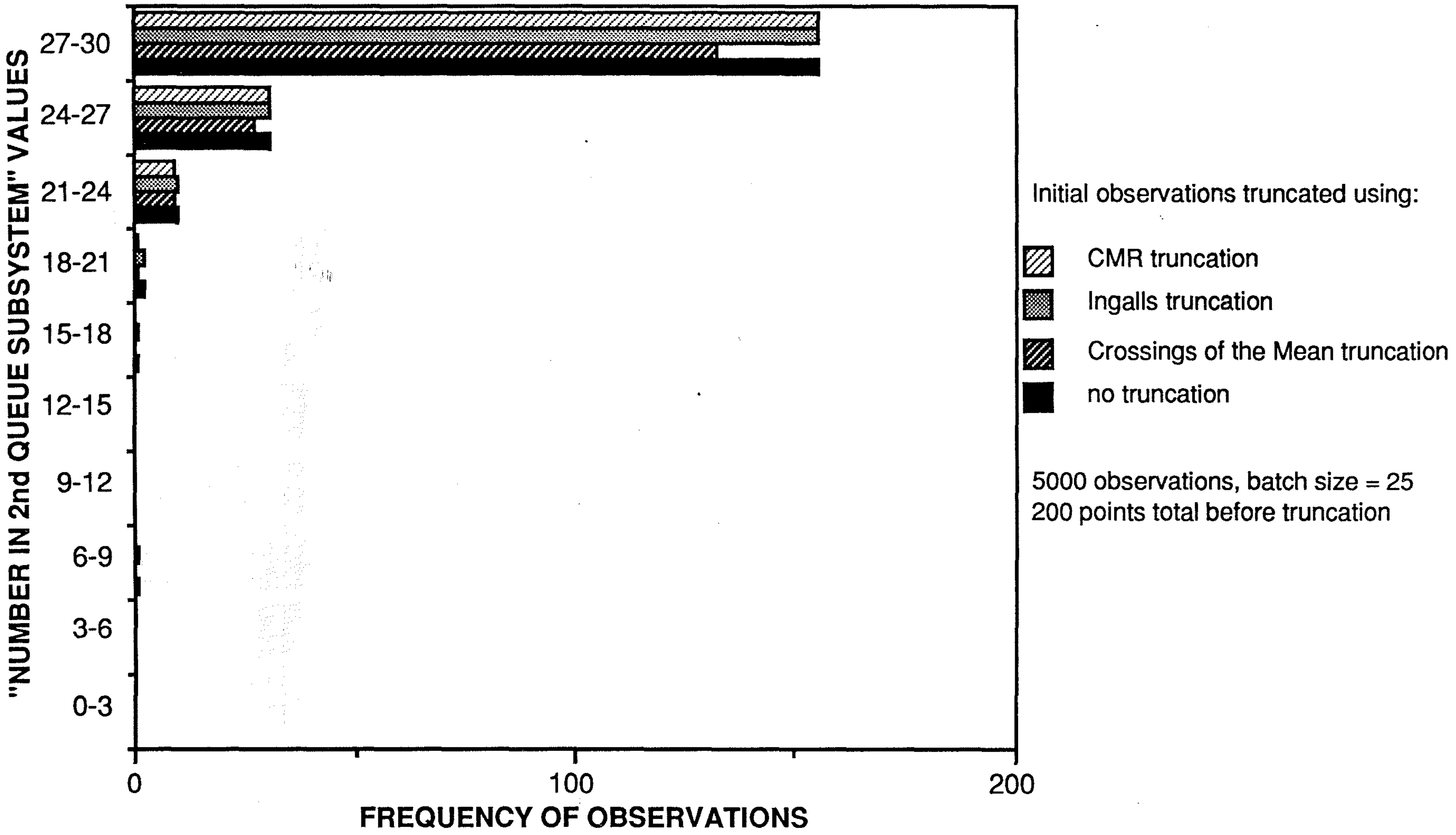
Queue Series Model: Run 1
Frequency of Batched Observations Within
Each Value Range



5000 observations, batch size = 25
200 points total before truncation

Figure 4.15

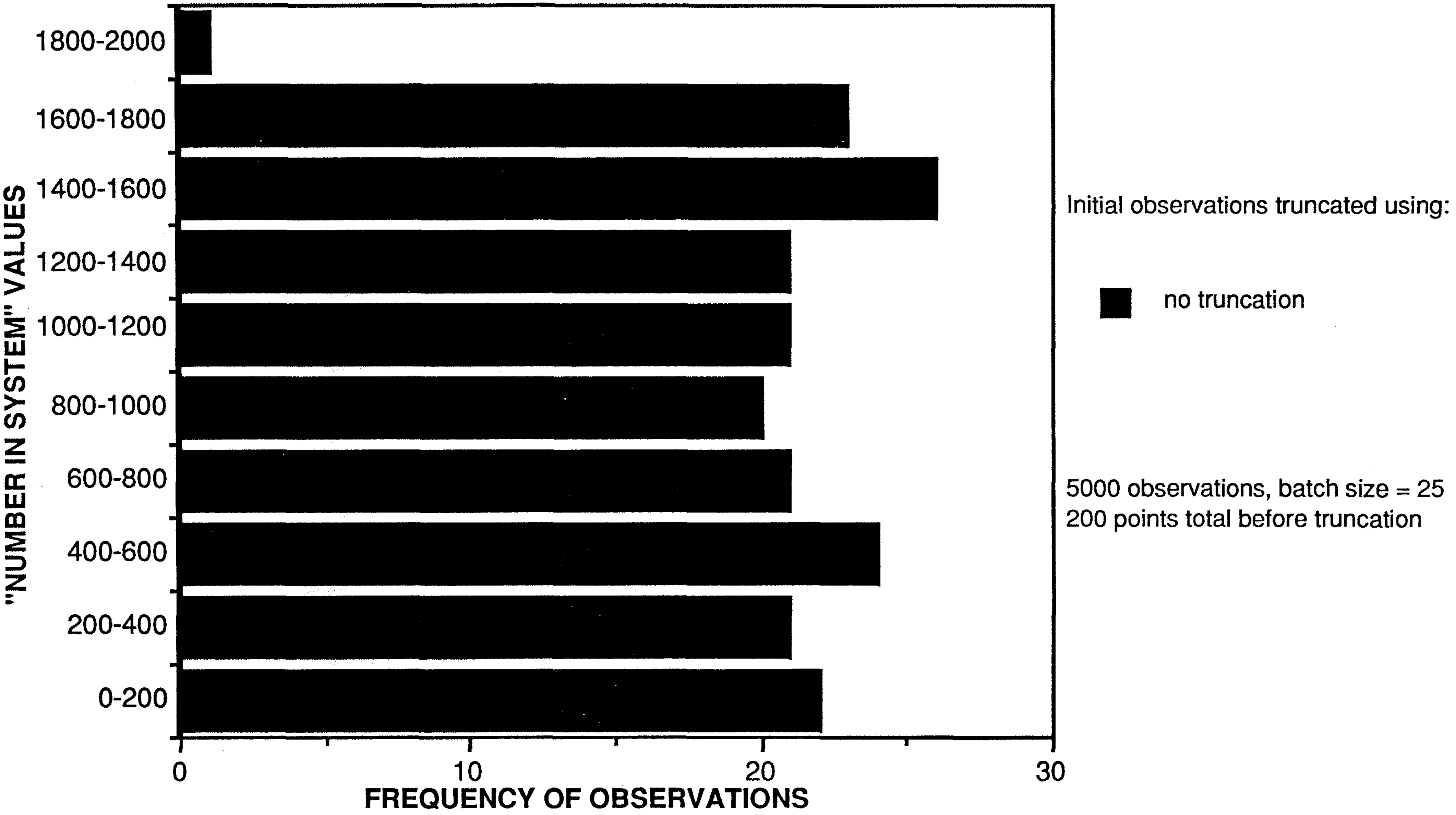
Filling Queue Model: Run 1
Frequency of Batched Observations Within
Each Value Range



5000 observations, batch size = 25
200 points total before truncation

Figure 4.16

Transient Queue Model: Run 1
Frequency of Batched Observations Within Each Value Range



5000 observations, batch size = 25
200 points total before truncation

Figure 4.17

graph is presented here simply to show its entirely transient behavior in another form. For the remaining models (the Loaded Queue and Filling Queue models), CMR truncation eliminates outlying data values that result from the biased initial portions of the runs. CMR truncation of the Loaded Queue output "smooths" the distribution by removing the unreasonably large "number in system" observations from the distribution, while for the Filling Queue output, it removes the unreasonably small "number in second queue subsystem" observations. For the Empty Queue, Queue Series, and Filling Queue models, the Cross/Mean and Ingalls heuristics call for a larger number of observations to be truncated than the CMR. The problem with this is that truncation beyond the CMR point does not significantly change the shape of the frequency curve. It only decreases the frequency of points in the steady-state value ranges, thereby reducing the confidence of the statistics. Hence, truncation beyond the CMR point is inefficient because its main effect is to eliminate steady-state data.

4.5 TESTING CONCLUSIONS

This chapter showed, through a variety of tests, that the CMR is an effective, although not perfect, approach to initialization bias detection and truncation. It also illustrated how confidence maximization can be used to compare the effectiveness of detection heuristics. Finally, it was shown that the CMR is a more effective detection heuristic than two previously developed heuristics, the Crossings-of-the-Mean Rule, and Ingalls' Algorithm.

CHAPTER FIVE
CONCLUSIONS

5.1 SUMMARY AND CONCLUSIONS

The purpose of this project was to evaluate the performance of various steady-state detection methodologies, in the light of new ideas about steady state and system settling time for discrete event dynamic systems. There have been several heuristics evaluation methodologies developed prior to this study, but almost all of them have relied on infinite samples and/or theoretical statistics to assess performance. Because of the emphasis on theoretical statistics, these methodologies are often meaningless in the context of the finite sample DEDS that are studied in "real world" simulations.

In order to develop a new, more meaningful steady-state detection methodology, three previous research efforts were studied in detail: Gafarian *et al.* (1978), Wilson and Pritsker (1978a,b), and Schruben (1981)/Heidelberger and Welch (1983). The work of Gafarian *et al.* was the first major attempt to set up a methodology by which to evaluate the effectiveness of steady-state detection heuristics. Their methodology was used to evaluate several longstanding "rule-of-thumb" procedures that have been used for steady-state detection. Their conclusion that none of these heuristics is acceptable was both unreasonably rigid and unproductive. It was overly rigid in that the heuristic-based statistics of each model were compared against its theoretical statistics as the measure of the heuristic's validity. It was unproductive because it gave no indication as to what sort of detection heuristic might be more acceptable than those tested.

Wilson and Pritsker's study was more theoretically sound than Gafarian *et al.'s*. They developed an approach to evaluating "initialization policies", which are combinations of initial conditions and detection heuristics. Their approach is very logical and is based on finite sample statistics (unlike Gafarian *et al.'s*), but in actual application, their methodology leaves something to be desired. To use their approach in evaluating several initialization policies for several model types would be quite cumbersome because of the need for tables of data based on probability transition functions for each output set. In addition, their conclusion that the best initialization policy is to set the initial conditions strategically and truncate no observations at all is, in most cases, unhelpful. The simulationist is often unaware of the "correct" initial conditions and must, therefore, rely on truncation of initialization bias to clean up his output data.

Heidelberger and Welch's run length control procedure, which uses Schruben's transient detection methodology, is an effective approach to locating the optimal truncation point. The principal drawback to their approach is the computational intensity of Schruben's procedure; however, as part of a software-based output analysis system (as the procedure is meant to be), Heidelberger and Welch's methodology is quite sound.

Based on lessons learned from these prior research efforts, the approach taken in this project has been to use finite samples and empirically-developed statistics as the basis for the heuristic evaluation. With such an approach, the results can be applied to finite sample simulation output that is not analytically tractable (e.g., through queuing theory or other straightforward mathematical techniques). New definitions of steady state and settling time for DEDS that use finite sample confidence level comparisons were conceived. A new steady-state detection methodology was created from these definitions that uses finite sample confidence statistics to determine the optimal truncation point for removal of initialization bias. With this methodology, the confidence statistics generated for a given (finite) sample or set of samples using various truncation heuristics are compared with one another to evaluate the relative performance of the heuristics.

Although it was initially conceived as a methodology for evaluating the performance of existing detection heuristics, once testing began, it became apparent that the new confidence level comparison concept could be used as a steady-state detection and initialization bias truncation procedure in itself. It also became clear that, as a detection procedure (the Confidence Maximization Rule), the new approach was more effective than most of the existing heuristics. A series of tests was conducted, in which the CMR was compared with two seemingly strong detection heuristics: the Crossings-of-the-Mean Rule and a recently developed algorithm by Ingalls. The heuristics were compared for five different types of DEDS: an initially empty and idle M/M/1 queue ($\rho = 0.9$), a heavily front-loaded M/M/1 queue ($\rho = 0.9$), a system of 15 M/M/1/15 queues in tandem ($\rho = 0.9$), a system of two parallel M/M/1/30 queues ($\rho_1 = 25$, $\rho_2 = 22.5$), and a transient system of two parallel M/M/1 (unconstrained) queues ($\rho_1 = 25$, $\rho_2 = 22.5$). The results of these tests indicate that, in general, the CMR outperforms the Crossings-of-the-Mean Rule and Ingalls' algorithm in both accuracy (i.e., high confidence and high coverage) and efficiency (i.e., fewer observations truncated). All three heuristics were equally effective at identifying the lack of a steady state in the transient model output.

Because of sample size/simulation length restrictions, steady-state detection and initialization bias truncation is often a more practical approach than dilution of initialization bias. In this project, the confidence level performance and reliability of "extended run" samples was tested to provide additional information to the reader; however, the main focus was on the comparison of detection and bias truncation heuristics and methodologies. A comparison of the statistics generated from the CMR against those from dilution shows that dilution using extremely large samples produces slightly better statistics than CMR truncation using significantly smaller samples. (For this test, the diluted samples were six times larger than the truncated samples.) It is important to note that the degree of improvement in confidence and coverage obtained by the use of dilution is quite small in comparison with the associated increase in sample size. Therefore, for most "real life" applications, in which sample size/simulation length is a primary constraint, truncation of initialization bias, requiring relatively small samples, is more effective than dilution of initialization bias, which requires relatively large samples.

The trade-off between computation time added by using the CMR and time required to generate additional data is clearly an issue here. In its current state (i.e., manual computations) the CMR can take some time to use, particularly if the estimated size of the transient region is unknown. However, the complexity of the computations required is low, and the process of using the CMR is logically straightforward enough that it should be rather simple to program. Once the procedure is automated, the computation time will probably be significantly smaller than the amount of time required to generate a "statistically" equivalent number of observations.

Even for cases in which statistical accuracy is more critical than sample size constraints, the use of truncation can improve results. Although in this test dilution of initialization bias, by increasing sample size, produced better statistical results than truncation, even better results can be obtained by using both an extended sample size *and* truncation of initialization bias.

The tests described in this paper have shown that, for general purpose, "real life" non-terminating simulation output analysis, an effective way to handle initialization bias is to use the CMR to identify and truncate the bias before system statistics are calculated. Although it does not perform perfectly in all cases, it seems to produce the best results, in terms of reliability, consistency, and efficiency, for a variety of DEDS types. In addition, the same confidence maximization concept can be used to compare the performance of other detection/truncation heuristics and methods for treating initialization bias.

5.2 RECOMMENDATIONS

This project has been an attempt to extend existing ideas about steady-state detection so as to be more applicable to the needs of "real world" simulation output analysis. The result of this project is a new foundation of concepts that use finite sample confidence levels to evaluate simulation output and determine optimal system statistics. From this foundation, a great deal remains to be done. There are many unresolved issues associated with the confidence maximization concept and steady-state detection, in general, that remain. Some of these will be addressed in this final section.

One area associated with the CMR that requires further exploration is reliability and consistency testing. The testing done in this project has given an indication as to the effectiveness of the CMR in detecting initialization bias and optimizing sample statistics, but additional testing must be done in a less software- and hardware-restrictive environment. The tests performed for this research were done on an 80286-based PC running at 8 MHz under DOS. Rapidly evolving microcomputer technology now permits more robust testing because longer runs (hence, larger and more batches) can be used for statistical calculations and comparisons. In addition, with more powerful software and hardware, a larger number of samples and a larger variety of DEDS models can be studied.

Another area requiring further work is the method of testing used in CMR evaluation. In order to broaden the test base of the CMR, it should also be evaluated using an entirely different heuristic evaluation methodology. One way to approach this would be to use a modified version of Wilson and Pritsker's heuristics' evaluation methodology (described in Chapter Two). A few changes would help to make their methodology more appropriate for this particular application. For example, their comparison of initialization policies includes the strategic setting of initial conditions. In testing the CMR, setting of initial conditions should not be included because the comparisons of interest are specifically among truncation methods. In addition, their methodology should be expanded to include testing under a wider variety of DEDS models because, as was shown in Chapter Four, heuristics can tend to work better for some model output types than for others. With these modifications, testing the CMR using Wilson and Pritsker's methodology will help to confirm the CMR's overall effectiveness.

Another area that must be studied more fully is the problem encountered with the Loaded Queue Model in Chapter Four, involving an initial transient that is so long and seemingly stable that the CMR identifies it as a steady state. The potential enhancement to the CMR, described in Chapter Four, that addresses this problem should be explored and tested.

One final area that must be explored further is the application of the CMR to a run length control procedure. The idea of ultimately using this initialization bias detection and truncation method as a part of an overall output analysis system has been a primary goal in this project. Although out of the scope of the work done here, extending the use of the truncation method to help determine appropriate simulation run lengths is the next logical step. Since the CMR requires calculations of confidence statistics, the most significant groundwork for a run length control procedure has already been laid. What is still needed is a methodical process, in which overall statistical accuracy requirements are specified at the start, the CMR is used to optimize the statistics for a given sample, and the sample statistics are compared with the requirements. If the sample statistics do not measure up, the sample being tested must be extended and retested until the requirements are met. The run length control procedure outlined in the paper by Heidelberger and Welch (1983), described in Chapter Two of this report, might work quite well alongside the CMR. In their procedure, Schruben's method is used for transient detection and truncation. With some very slight modifications, the CMR might be incorporated into Heidelberger and Welch's procedure in place of Schruben's more complex and time-consuming method. Exploration and testing of this possibility would not be difficult and should be the next task in this area.

The issues listed in this section are a few of the most important areas of steady-state detection in which further work must be done. Once these issues have been explored more fully, steady-state detection, probably the most complex part of non-terminating simulation output analysis, will be better understood. At that point, effective automation of the output analysis process will be within reach.

REFERENCES

Conway, R. W. (1963), "Some Tactical Problems in Digital Simulation", Management Science, vol. 10, pp. 47-61.

Daniel, Wayne W. (1977), Introductory Statistics with Applications, Boston, MA: Houghton Mifflin Co.

Devore, Jay L. (1982), Probability and Statistics for Engineering and the Sciences, Monterey, California: Brooks/Cole Publishing Co.

Emshoff, James R. and Roger L. Sisson (1970), Design and Use of Computer Simulation Models, U.S.A.: MacMillan Co.

Fishman, George S. (July/August 1972), "Bias Considerations in Simulation Experiments", Operations Research, vol. 20, no. 4.

Fishman, George S. (1973), Concepts and Methods in Discrete Event Digital Simulation, New York: John Wiley & Sons, Inc.

Fishman, George S. (1978), Principles of Discrete Event Simulation, New York: John Wiley & Sons, Inc.

Gabbert, Paula and Russell Sharples (1986), "Formalization of an Expert System for Simulation Output Analysis", University of Virginia, Department of Systems Engineering, Independent Study.

Gafarian, A. V., C. J. Ancker, Jr., and T. Morisaku (1978), " Evaluation of Commonly Used Rules for Detecting 'Steady State' in Computer Simulation", Naval Research Logistics Quarterly, vol 25, pp. 511-529.

Gordon, G. (1969), System Simulation, New Jersey: Prentice-Hall.

Heidelberger, Philip and Peter D. Welch (1983), "Simulation Run Length Control in the Presence of an Initial Transient", Operations Research, vol. 31, no. 6, pp. 1109-1144.

Ingalls, R. (1987), personal correspondence.

Law, Averill M. and W. David Kelton (1982), Simulation Modeling and Analysis, USA: McGraw-Hill, Inc.

Naylor, Thomas E. (1969), The Design of Computer Simulation Experiments, Durham, North Carolina: Duke University Press.

O'Keefe, Robert (January 1986), "Simulation and Expert Systems: A Taxonomy and Some Examples", Simulation, vol. 46, no. 1, pp. 10-16.

Papoulis, Athanasios (1965), Probability, Random Variables, and Stochastic Processes, USA: McGraw-Hill, Inc.

Pegden, C. Dennis (1986), Introduction to SIMAN, State College, Pennsylvania: Sysems Modeling Corporation.

Phillips, D. T., A. Ravindran, and J. J. Solberg (1976), Operations Research: Principles and Practice, USA: John Wiley & Sons, Inc.

Ross, Sheldon M. (1985), Introduction to Probability Models, Orlando, Florida: Academic Press, Inc.

Schribner, T. (1974), Simulation Using GPSS, New York: John Wiley & Sons.

Schruben, Lee W. (May/June 1982), "Detecting Initialization Bias in Simulation Output", Operations Research, vol. 30, no. 3, pp. 569-590.

Shannon, R. E. (1975), System Simulation: The Art and Science, Englewood Cliffs, New Jersey: Prentice-Hall.

Shannon, R. E. (1984), "Artificial Intelligence and Simulation", Proceedings of 1984 Winter Simulation Conference, pp. 3-10.

Soloman, Susan L. (1983), Simulation of Waiting-Line Systems, Prentice-Hall.

Wilson, J. R. and A. A. B. Pritsker (August 1978), "A Survey of Research on the Simulation Startup Problem", Simulation, vol. 31, no. 2, pp. 55-58.

Wilson J. R. and A. A. B. Pritsker (September 1978), "Evaluation of Startup Policies in Simulation Experiments", Simulation, vol. 31, no. 3, pp. 79-89.

APPENDIX A

SIMAN MODEL AND EXPERIMENT FILES FOR TEST SYSTEMS

I. Empty Queue Model File

```
BEGIN;
          CREATE:ED(1);
          COUNT:1;
          COUNT:2;
          QUEUE,1;
          SEIZE:SERVER;
          DELAY:ED(2);
          RELEASE:SERVER;
          COUNT:1,-1;
          COUNT:2:DISPOSE;
END;
```

Experiment File

```
BEGIN;
DISCRETE,1000,,1;
RESOURCES:1,SERVER;
DISTRIBUTIONS: 1,EX(1,1):
               2,EX(2,1);
PARAMETERS: 1,10:
            2,9;
COUNTERS: 1,NUMBER IN SYSTEM,,,20:
          2,OBSERVATIONS,30000;
END;
```

II. Loaded Queue Model

```
BEGIN;
          CREATE,100;
          CREATE:ED(1);
CONT      COUNT:2;
          COUNT:1;
          QUEUE,1;
          SEIZE:SERVER;
          DELAY:ED(2);
          RELEASE:SERVER;
          COUNT:1,-1;
          COUNT:2:DISPOSE;
END;
```

Experiment File

```
BEGIN;
DISCRETE,1000,1,1;
RESOURCES:1,SERVER;
DISTRIBUTIONS: 1,EX(1,1):
               2,EX(2,1);
PARAMETERS: 1,10:
            2,9;
COUNTERS: 1,NUMBER IN SYSTEM,,,80:
          2,TOTAL OBSERV,30100;
END;
```

III. Queue Series Model File

```
BEGIN;
         CREATE:ED(1);
         ROUTE:0,1;

         STATION,1-14;
         BRANCH,1:
           IF,NC(M).GE.15,BALK:
           ELSE,CONT1;
CONT1    COUNT:M;
         QUEUE,M;
         SEIZE:SERVER(M);
         DELAY:ED(2);
         RELEASE:SERVER(M);
         COUNT:M,-1;
         ROUTE:0,M+1;

         STATION,15;
         BRANCH,1:
           IF,NC(15).GE.15,BALK:
           ELSE,CONT2;
CONT2    COUNT:M;
         COUNT:M+1;
         QUEUE,M,14;
         SEIZE:SERVER(M);
         DELAY:ED(2);
         RELEASE:SERVER(M);
         COUNT:M,-1;
         COUNT:M+1:DISPOSE;
BALK     COUNT:17:DISPOSE;
END
```

Queue Series Experiment File

```
BEGIN;
DISCRETE,2000,,15,15;
RESOURCES:1-15,SERVER;
DISTRIBUTIONS: 1,EX(1,1):
               2,EX(2,1);
PARAMETERS: 1,10:
            2,9;
COUNTERS: 1,NO.IN SUBSYS:
          2,NO.IN SUBSYS:
          3,NO.IN SUBSYS:
          4,NO.IN SUBSYS:
          5,NO.IN SUBSYS:
          6,NO.IN SUBSYS:
          7,NO.IN SUBSYS:
          8,NO.IN SUBSYS:
          9,NO.IN SUBSYS:
         10,NO.IN SUBSYS:
         11,NO.IN SUBSYS:
         12,NO.IN SUBSYS:
         13,NO.IN SUBSYS:
         14,NO.IN SUBSYS:
         15,NO.IN SUBSYS15,,,80:
         16,OBSERVATIONS,5000:
         17,NO. OF BALKS;
END;
```

IV. Filling (Constrained) Queue Network Model File

```
BEGIN;
         CREATE:ED(1);
         BRANCH,1:
            WITH,.3,SERVE1:
            ELSE,SERVE2;
SERVE1   BRANCH,1:
            IF,NC(1).GE.30,BALK:
            ELSE,CONT1;
CONT1    COUNT:1;
         COUNT:2;
         QUEUE,1,30;
         SEIZE:SERVER1;
         DELAY:ED(2);
         RELEASE:SERVER1:NEXT(LEAVE);
SERVE2   BRANCH,1:
            IF,NC(2).GE.30,BALK:
            ELSE,CONT2;
CONT2    COUNT:1;
         COUNT:2;
         QUEUE,2,30;
         SEIZE:SERVER2;
         DELAY:ED(3);
         RELEASE:SERVER2;
LEAVE    COUNT:1,-1;
         COUNT:2:DISPOSE;
BALK     COUNT:3:DISPOSE;
END;
```

Experiment File

```
BEGIN;
PROJECT,FILLUP,MAM,10/5/87;
DISCRETE,1000,1,2;
RESOURCES:1,SERVER1,5:
          2,SERVER2,6;
DISTRIBUTIONS:1,EX(1,1):
              2,EX(2,1):
              3,EX(3,1);
PARAMETERS:1,2:2,50:3,45;
COUNTERS: 1,NUM IN Q2,,,50:
          2,TOTAL OBS,30000:
          3,NUM BALKS;
;TRACE;
END;
```

V. Transient (Unconstrained) Queue Network Model File

```
BEGIN;
          CREATE:ED(1);
          BRANCH,1:
             WITH,.3,SERVE1:
             ELSE,SERVE2;
SERVE1    COUNT:1;
          COUNT:2;
          QUEUE,1;
          SEIZE:SERVER1;
          DELAY:ED(2);
          RELEASE:SERVER1:NEXT(LEAVE);
SERVE2    COUNT:1;
          COUNT:2;
          QUEUE,2;
          SEIZE:SERVER2;
          DELAY:ED(3);
          RELEASE:SERVER2;
LEAVE     COUNT:1,-1;
          COUNT:2:DISPOSE;
BALK      COUNT:3:DISPOSE;
END;
```

Experiment File

```
BEGIN;
PROJECT,FILLUP,MAM,10/5/87;
DISCRETE,2600,1,2;
RESOURCES:1,SERVER1,5:
          2,SERVER2,6;
DISTRIBUTIONS:1,EX(1,1):
              2,EX(2,1):
              3,EX(3,1);
PARAMETERS:1,2:2,50:3,45;
COUNTERS: 1,NUM IN Q2,,,60:
          2,TOTAL OBS,9000:
          3,NUM BALKS;
;TRACE;
END;
```

APPENDIX B

FORTRAN CODE FOR COMPILATION OF STATISTICS OVER A RANGE OF TRUNCATION POINTS

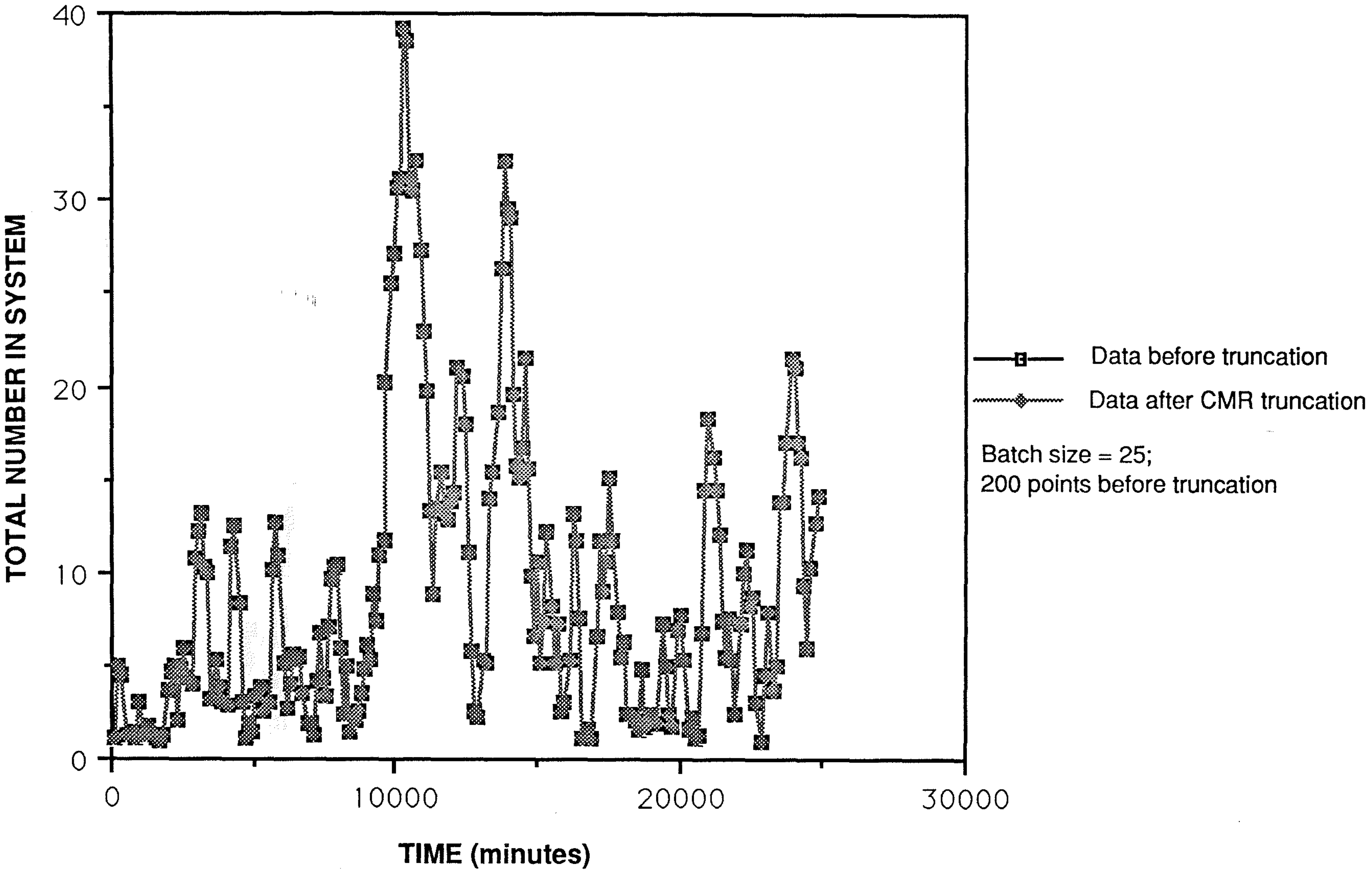
```
      PROGRAM STATISTICS
      REAL
MEAN,STDEV,VAR,DIFF,SUMDIFF,SQDIFF(5000),VALUE(5000),OBS,
     *TIME,TRTOTAL,SUM,VARMEAN,HW
      INTEGER COUNT,TRUNC,TOP
      TRUNC = 0
      SUM = 0.0
      COUNT = 0
      TOP = 0
      TRTOTAL = 0.0
      SUMDIFF = 0.0
      OPEN(1,FILE='OUTPUT.CUR')
      OPEN(2,FILE='STATS.TMP',STATUS='NEW')
      WRITE(2,5)
  5   FORMAT(3X,'PTS TRUN',7X,'MEAN',12X,'ST DEV',9X,'HW')
      READ(1,10)NTITLE1,NTITLE2
 10   FORMAT(1X,I6,1X,A20)
 20   READ(1,30)TIME,OBS
 30   FORMAT(1X,E14.8,1X,E14.8)
C     WRITE(*,12)TIME,STAT
 12   FORMAT(1X,E14.8,1X,E14.8)
      IF (TIME.NE.(-1.0)) THEN
         COUNT = COUNT + 1
         VALUE(COUNT) = OBS
         SUM = SUM + OBS
         GO TO 20
      ENDIF
      TOP = COUNT
C
      DO 50, TRUNC = 0,TOP,50
      SUMDIFF = 0.0
      TRTOTAL = 0.0
      IF (TRUNC.GT.0) THEN
C      WRITE(*,52)COUNT
 52   FORMAT(1X,'COUNT 1 IS ',I4)
          COUNT = TOP - TRUNC
          DO 51, K = (TRUNC-49),TRUNC
             TRTOTAL = TRTOTAL + VALUE(K)
 51       CONTINUE
          SUM = SUM - TRTOTAL
      ENDIF
      MEAN = SUM/COUNT
      DO 40, I = (TRUNC+1),TOP
C      WRITE(*,44)VALUE(I)
 44   FORMAT(1X,'VALUE ',F10.5)
         DIFF = VALUE(I) - MEAN
C      WRITE(*,41)TRUNC,TOP,DIFF,MEAN
 41   FORMAT(1X,I3,1X,I3,1X,F10.5,1X,F10.5)
         SQDIFF(I) = DIFF * DIFF
         SUMDIFF = SUMDIFF + SQDIFF(I)
C      WRITE(*,42)SQDIFF(I),SUMDIFF
 42   FORMAT(1X,F10.5,1X,'SUMDIFF ',F10.5)
 40   CONTINUE
C      WRITE(*,46)COUNT
 46   FORMAT(1X,'COUNT ',I4)
      VAR = SUMDIFF/(COUNT - 1)
      STDEV = SQRT(VAR)
      HW = 1.96*SQRT(VAR/COUNT)
C
      WRITE(2,60)TRUNC,MEAN,STDEV,HW
 60   FORMAT(5X,I4,5X,F11.4,5X,F10.4,5X,F10.4)
C
 50   CONTINUE
      STOP

END
```

APPENDIX C

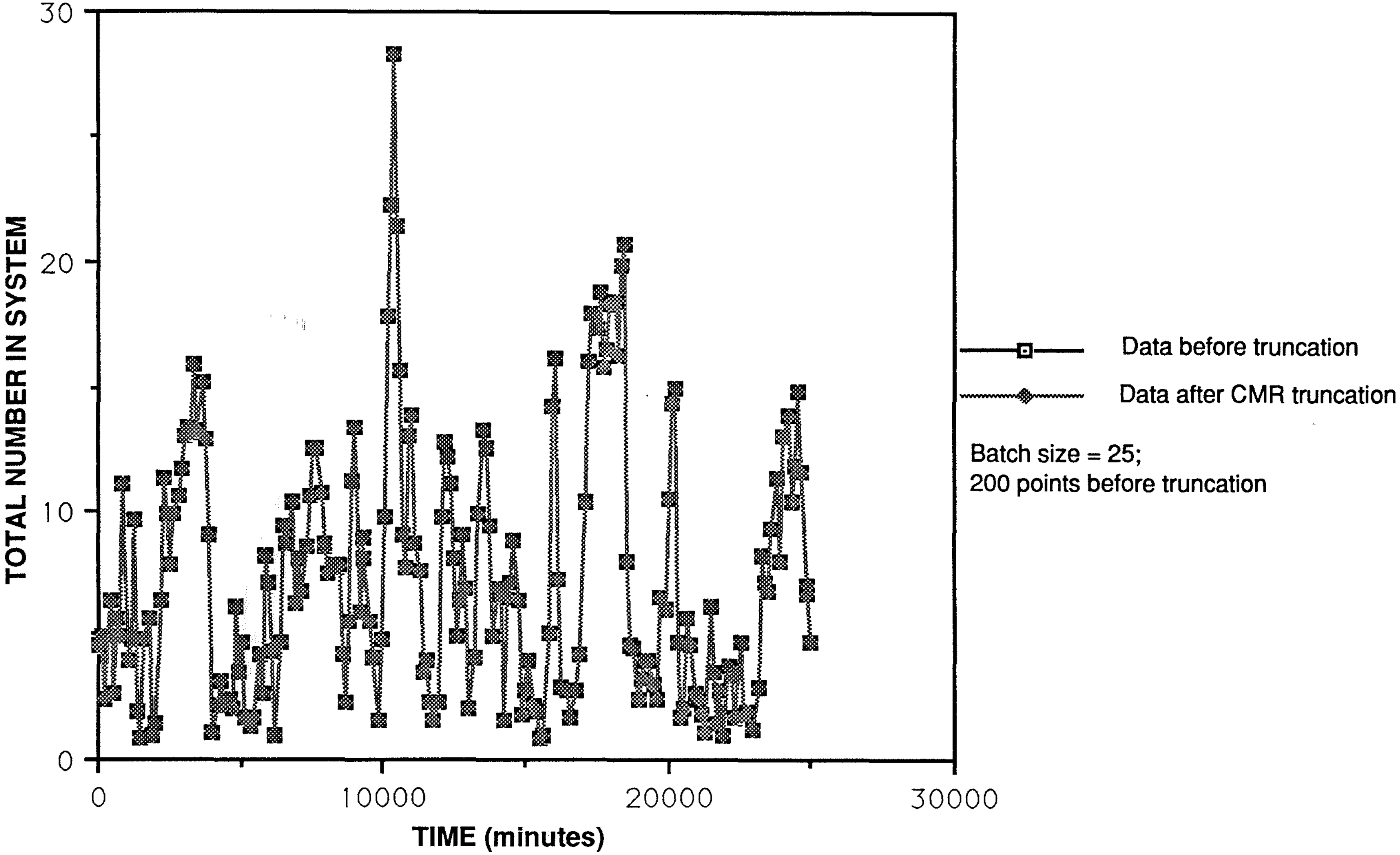
GRAPHS OF NINE REMAINING RUNS OF THE FIVE TEST MODELS
(5000 Observation Runs, Batched by 25)

Empty Queue Model : Run 2



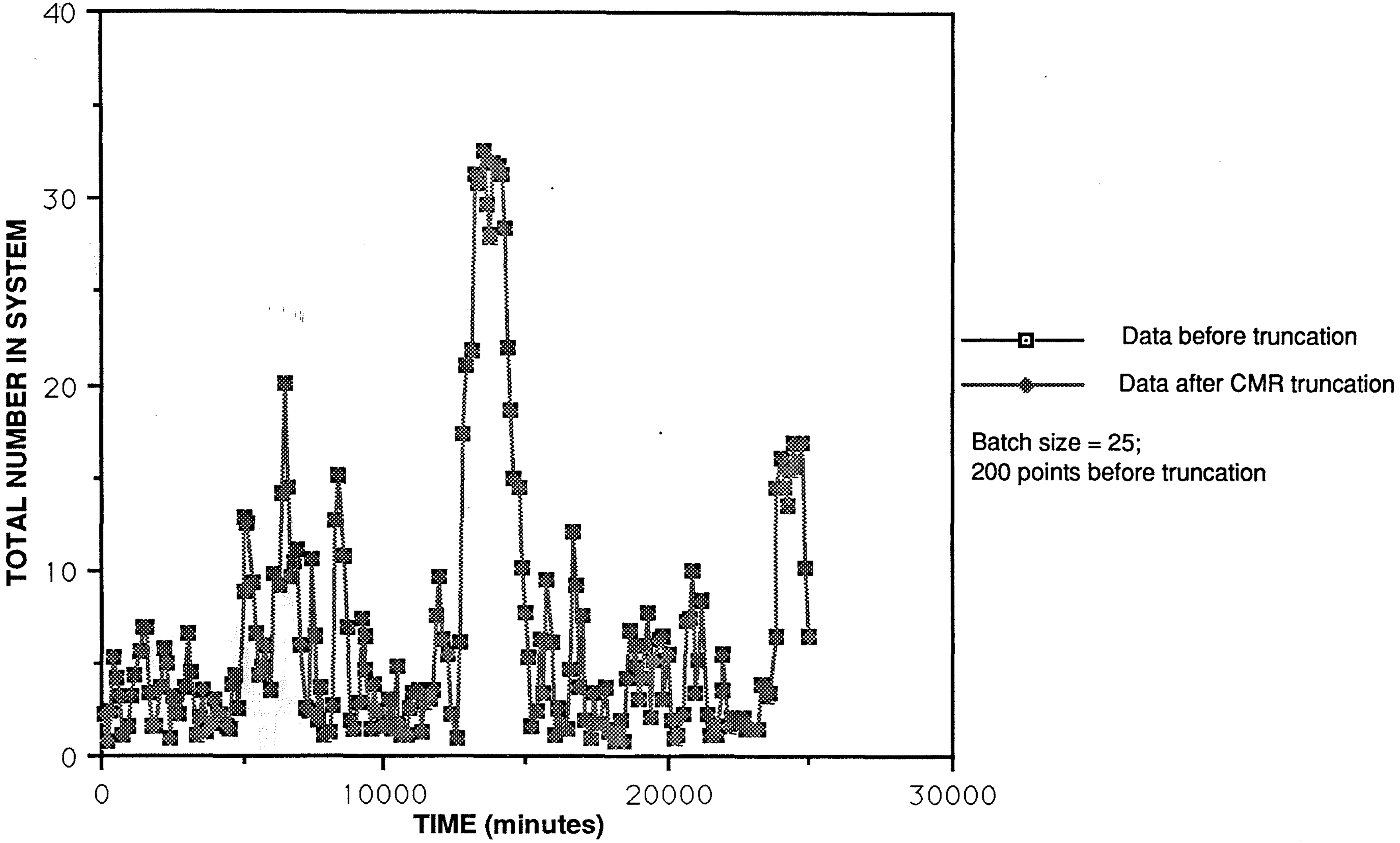
Batch size = 25;
200 points before truncation

Empty Queue Model : Run 3



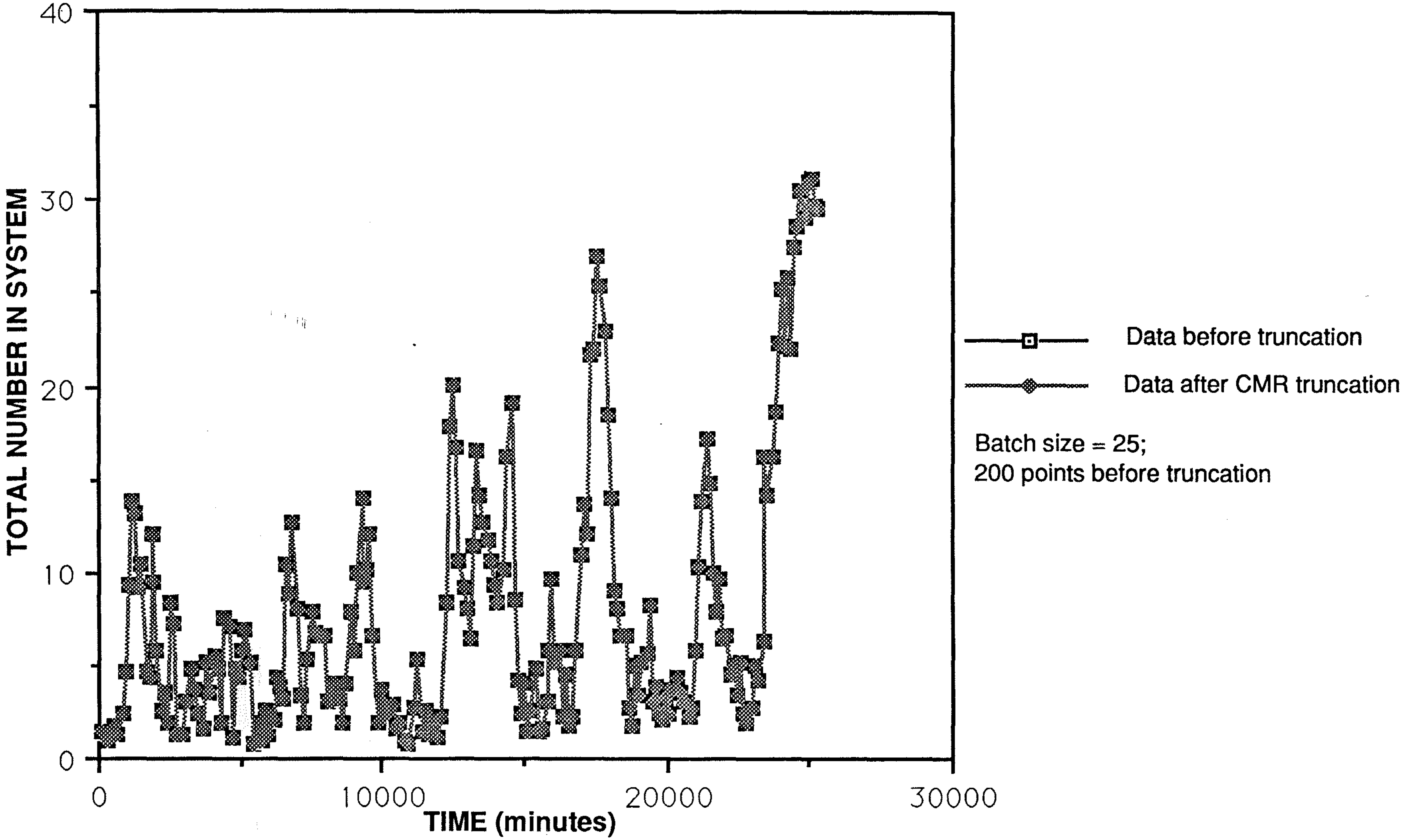
Batch size = 25;
200 points before truncation

Empty Queue Model : Run 4

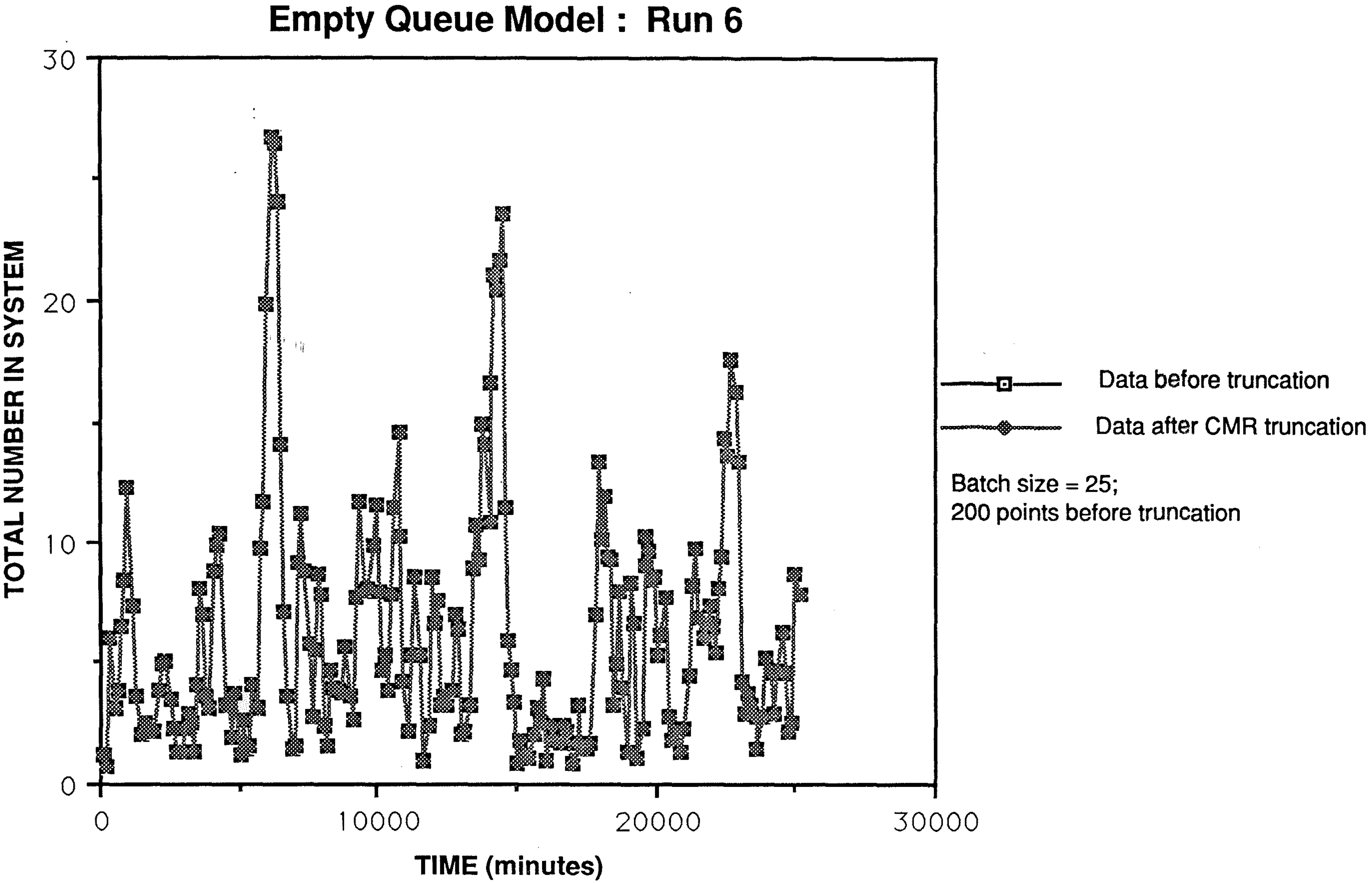


Batch size = 25;
200 points before truncation

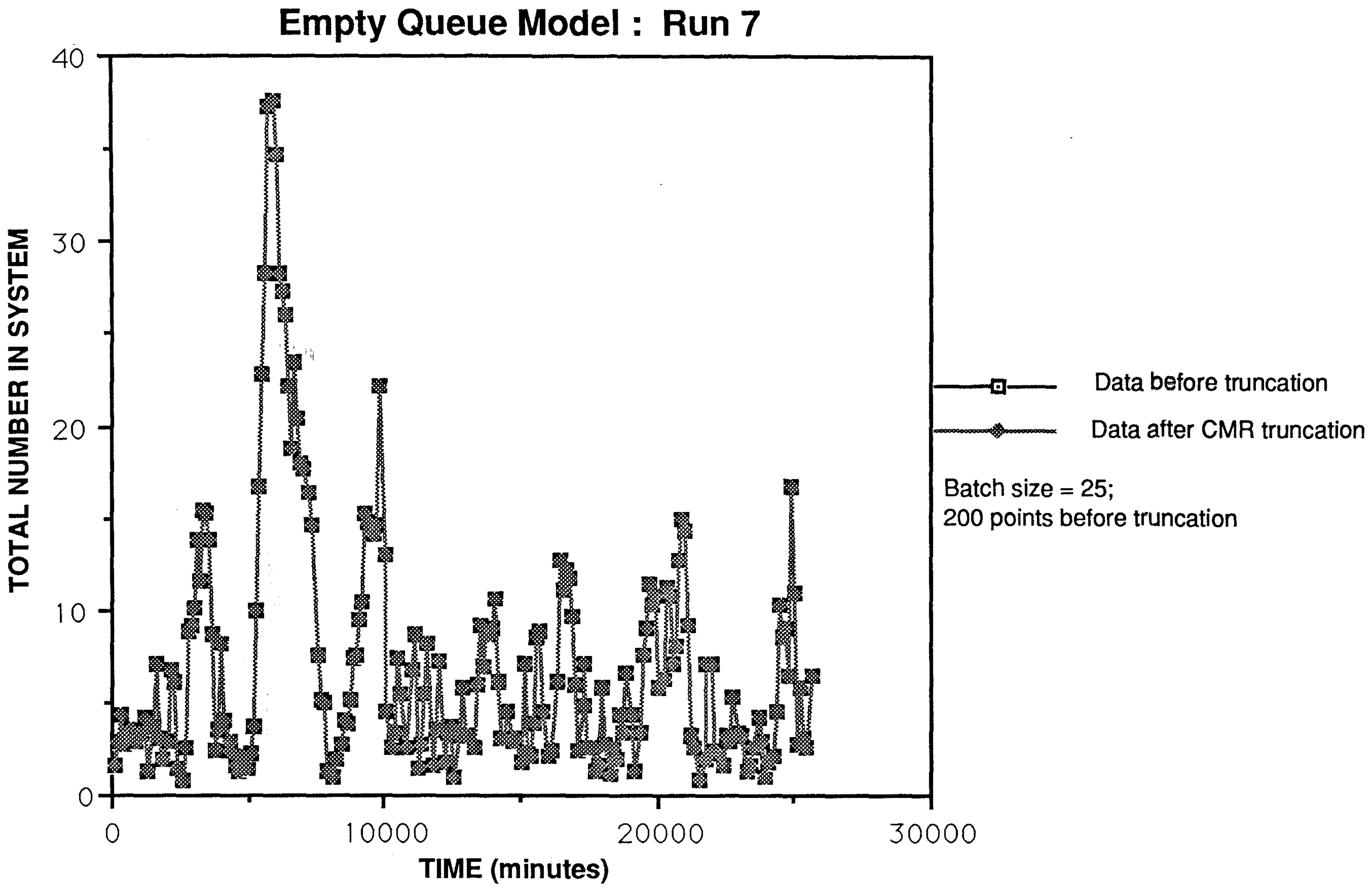
Empty Queue Model : Run 5



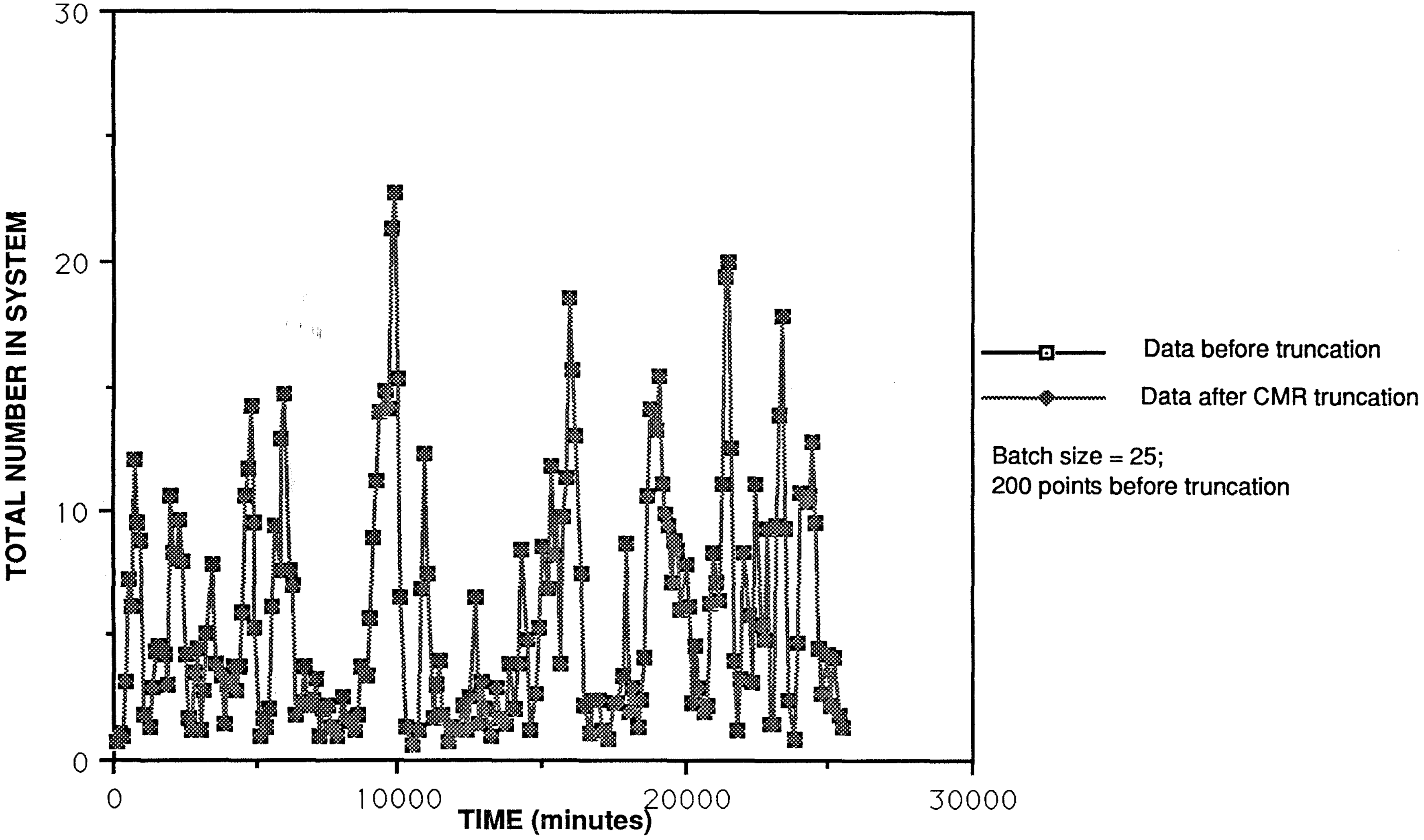
Batch size = 25;
200 points before truncation



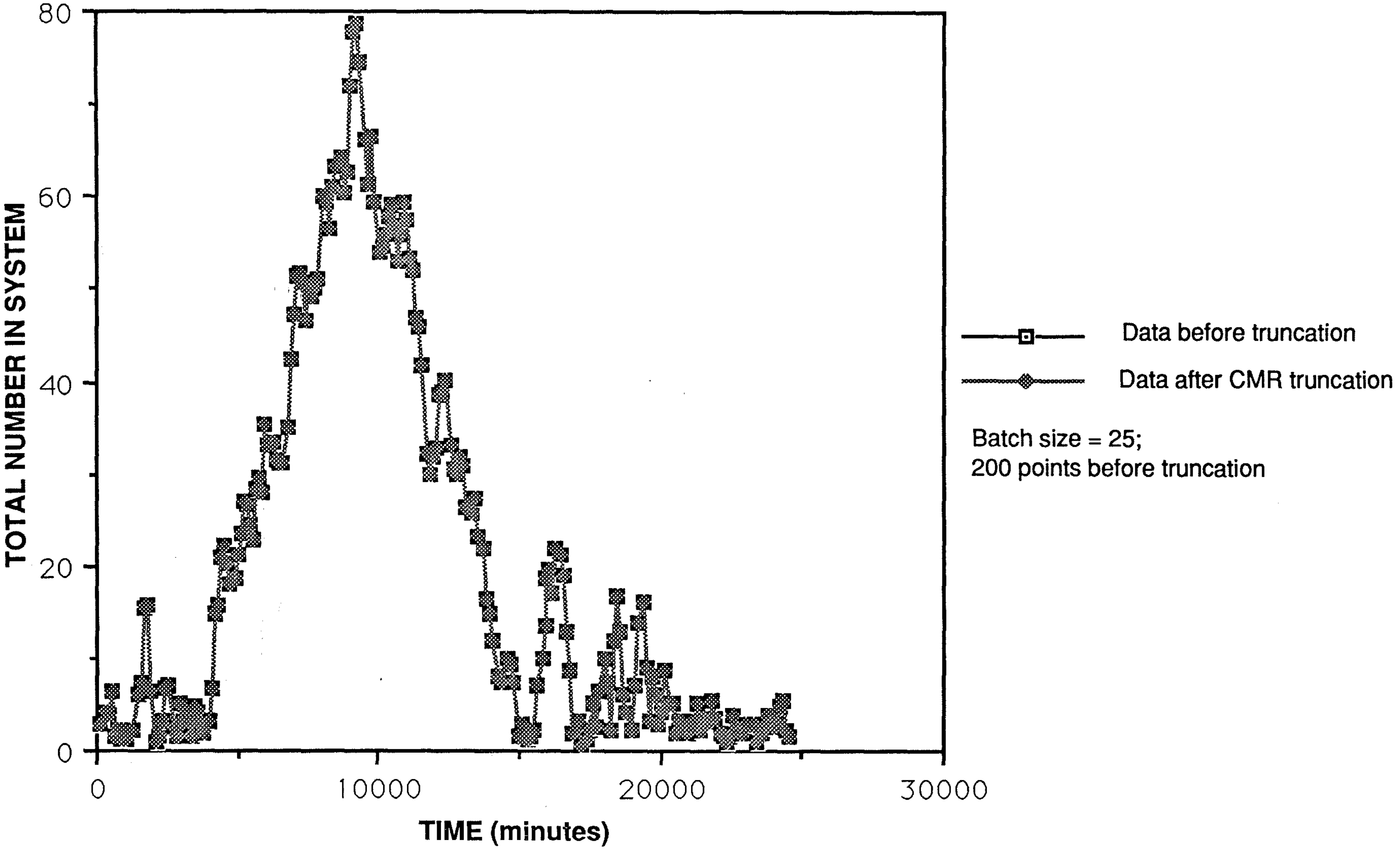
Batch size = 25;
200 points before truncation


Empty Queue Model : Run 7
TOTAL NUMBER IN SYSTEM
40
30
20
10
0
0
10000
20000
30000
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

Empty Queue Model : Run 8

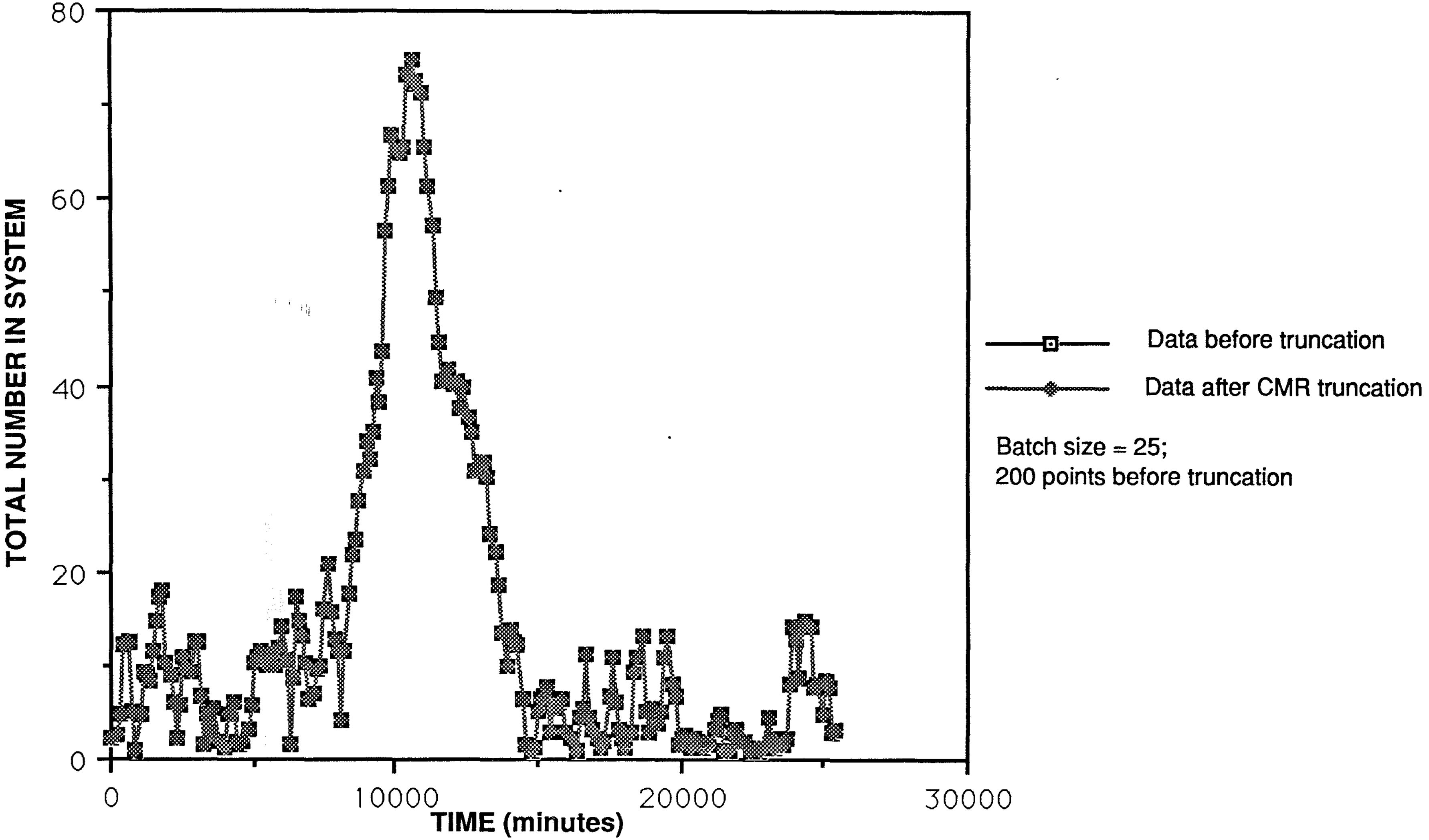


Empty Queue Model : Run 9

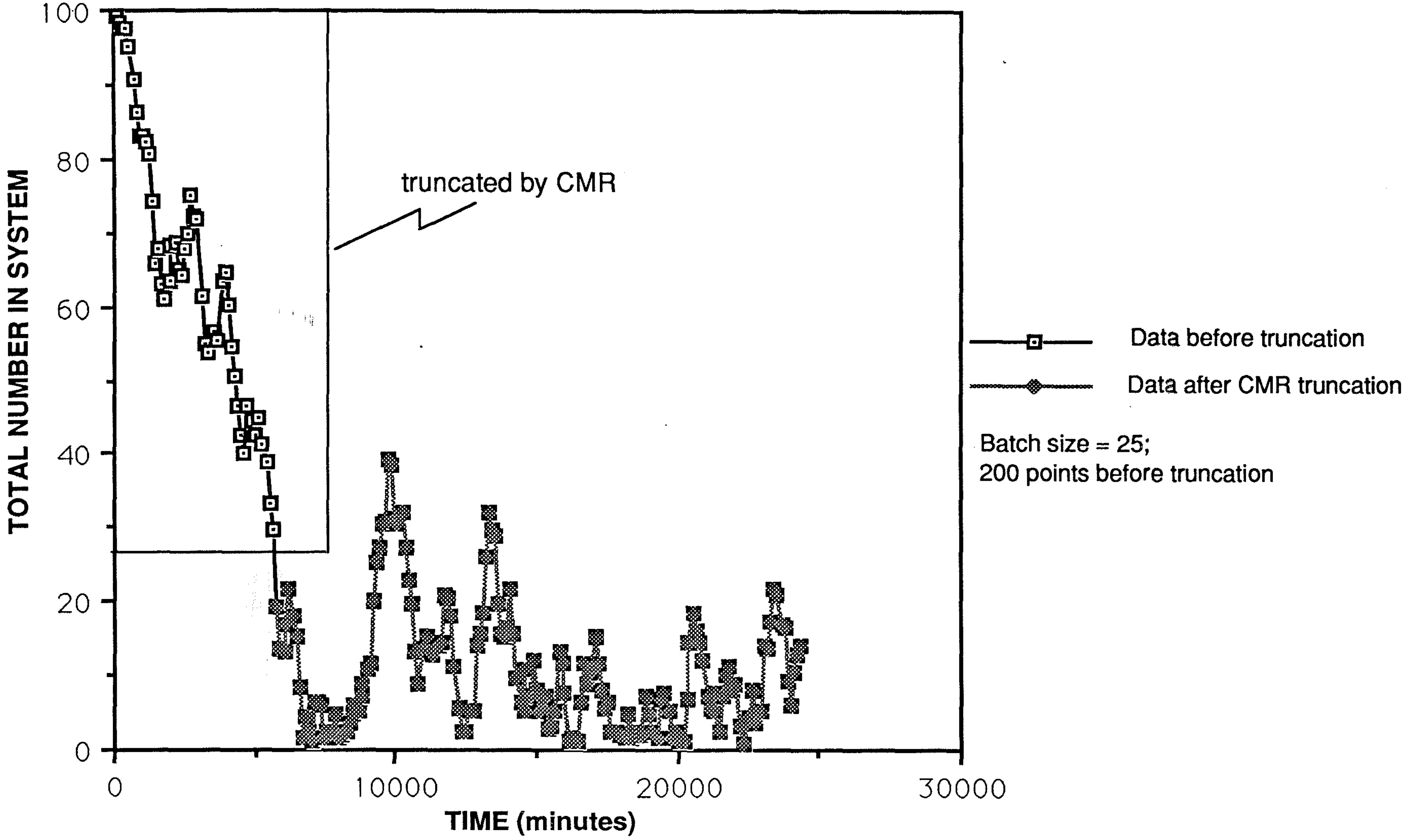


Batch size = 25;
200 points before truncation

Empty Queue Model : Run 10

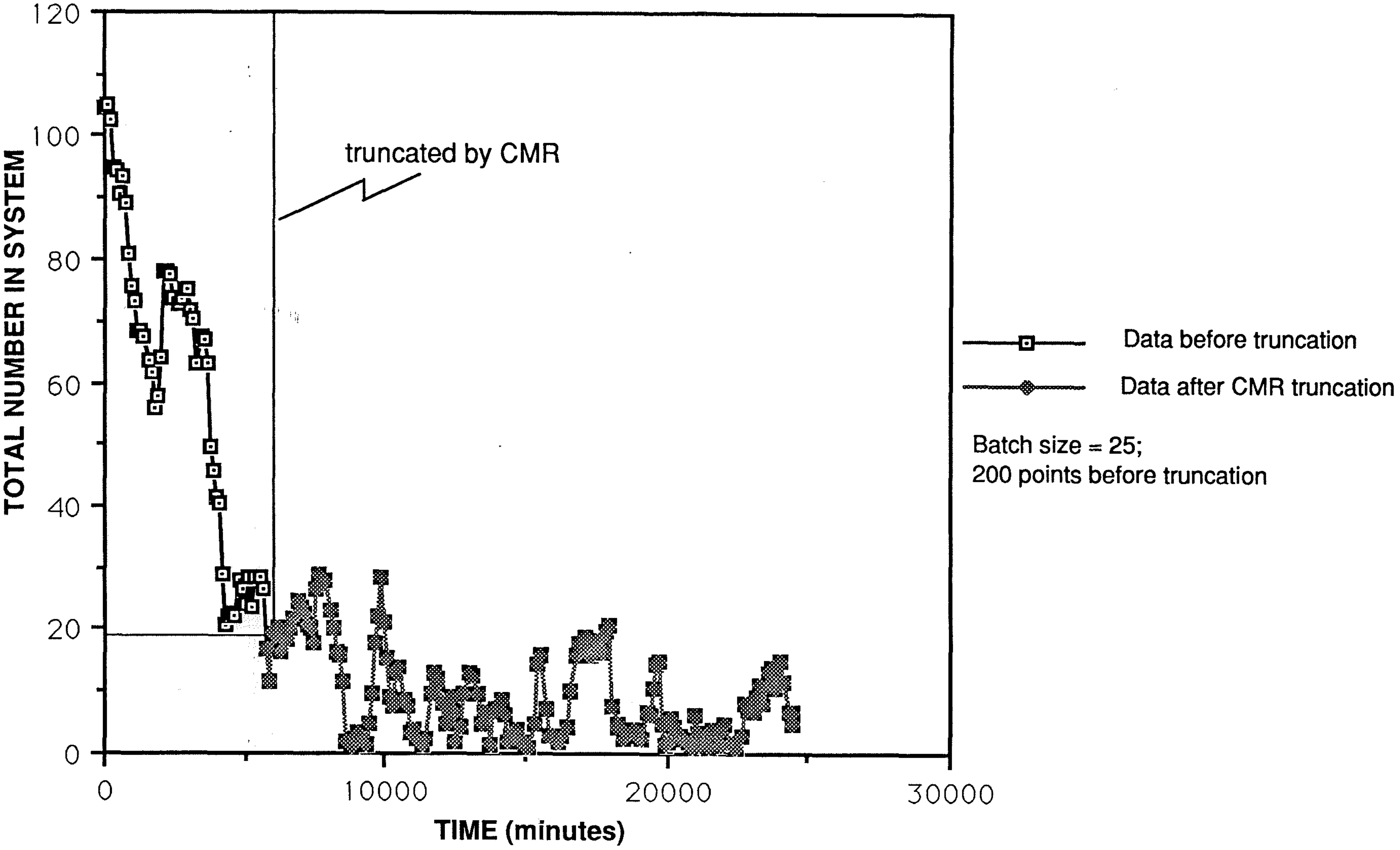


Loaded Queue Model : Run 2



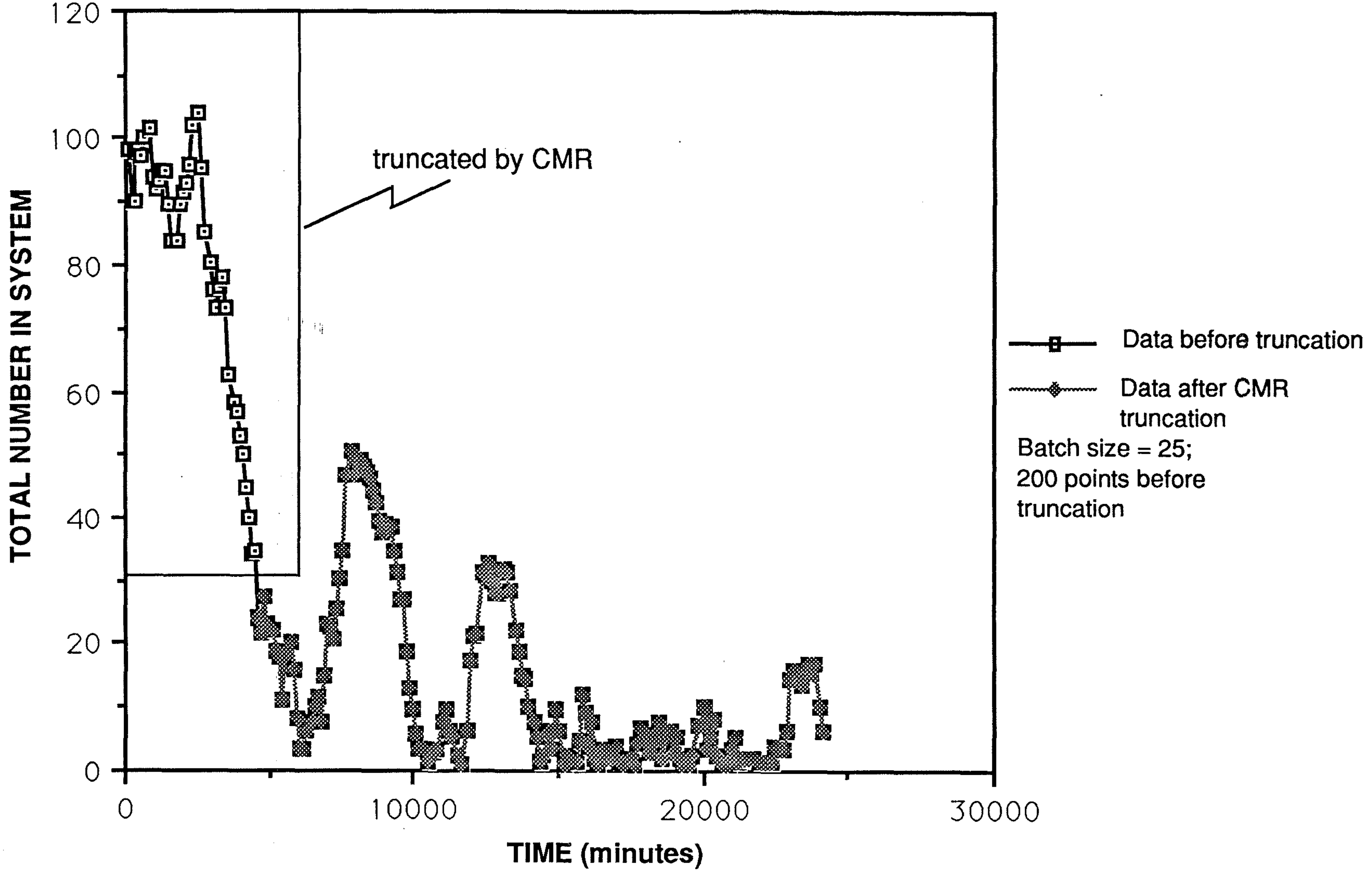
Batch size = 25;
200 points before truncation

Loaded Queue Model : Run 3

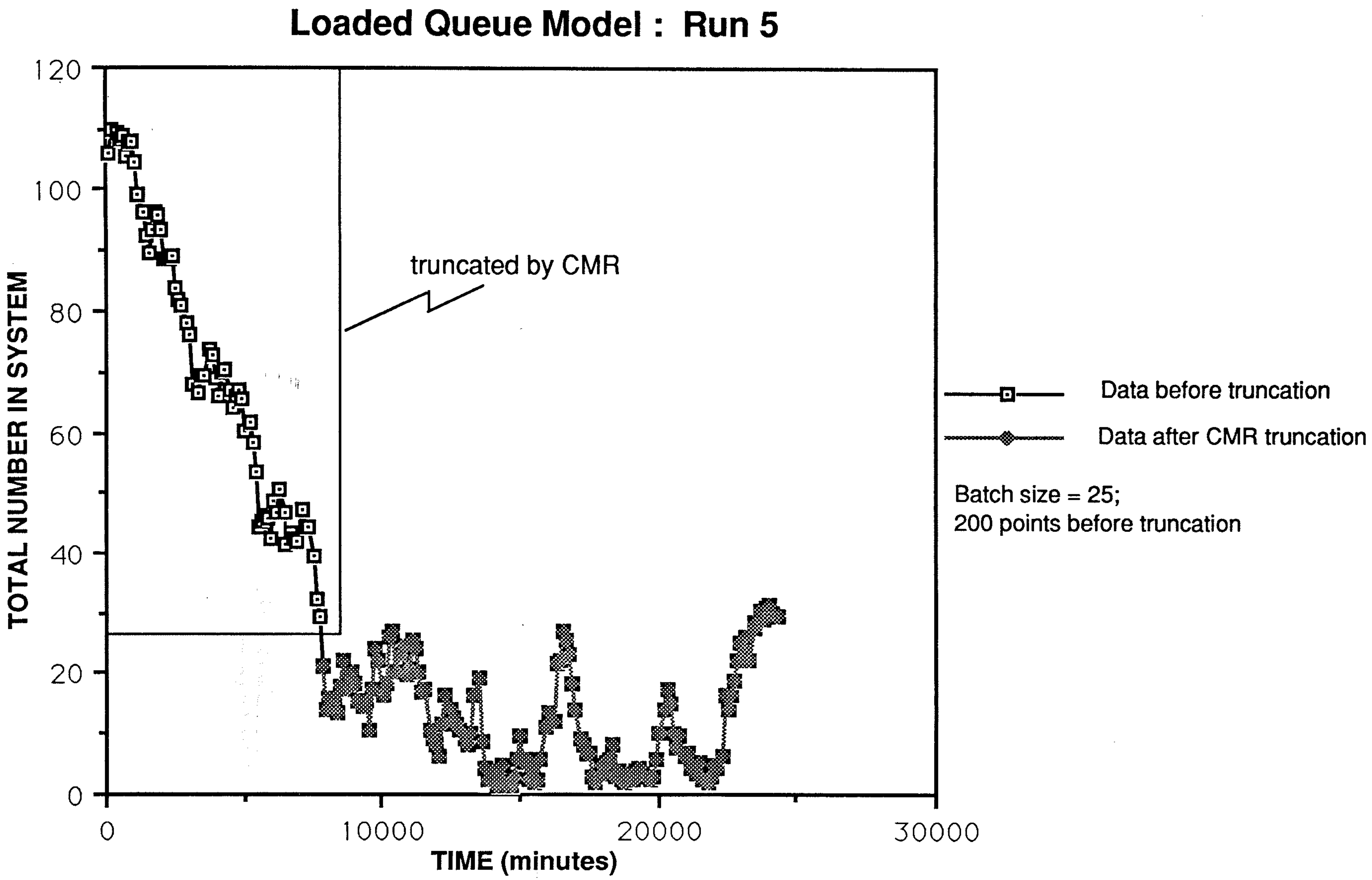


Batch size = 25;
200 points before truncation

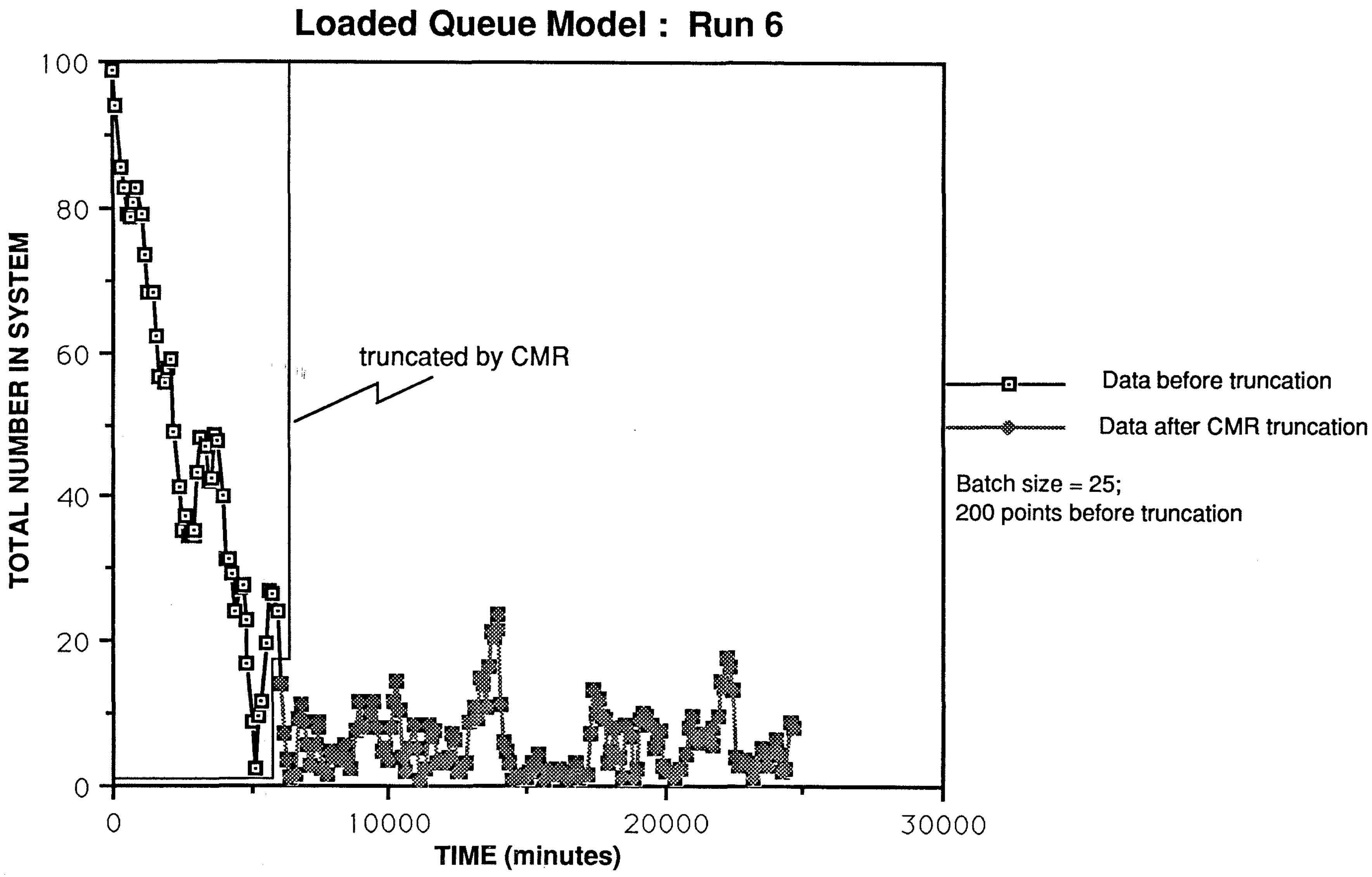
Loaded Queue Model : Run 4



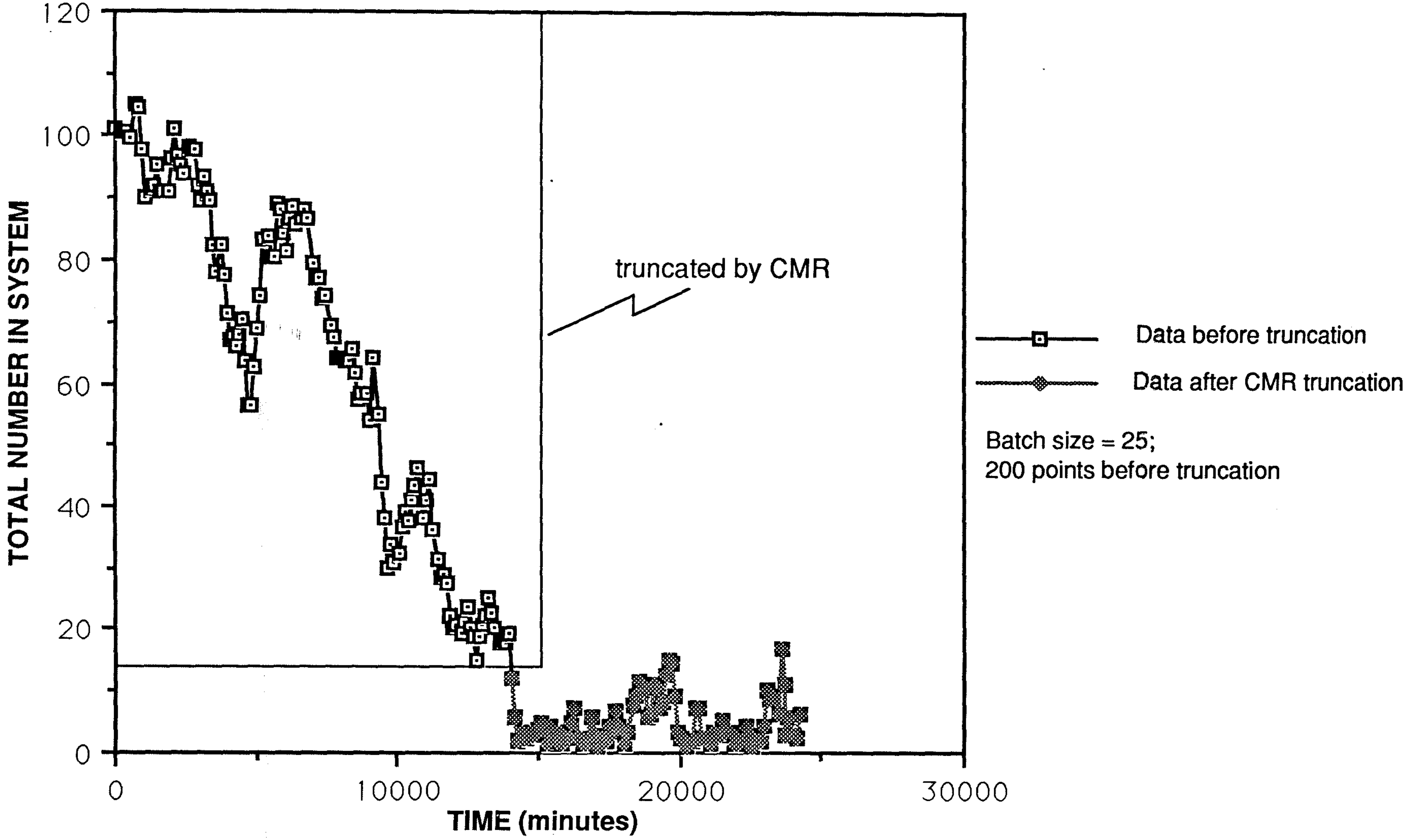
Batch size = 25;
200 points before truncation



Batch size = 25;
200 points before truncation

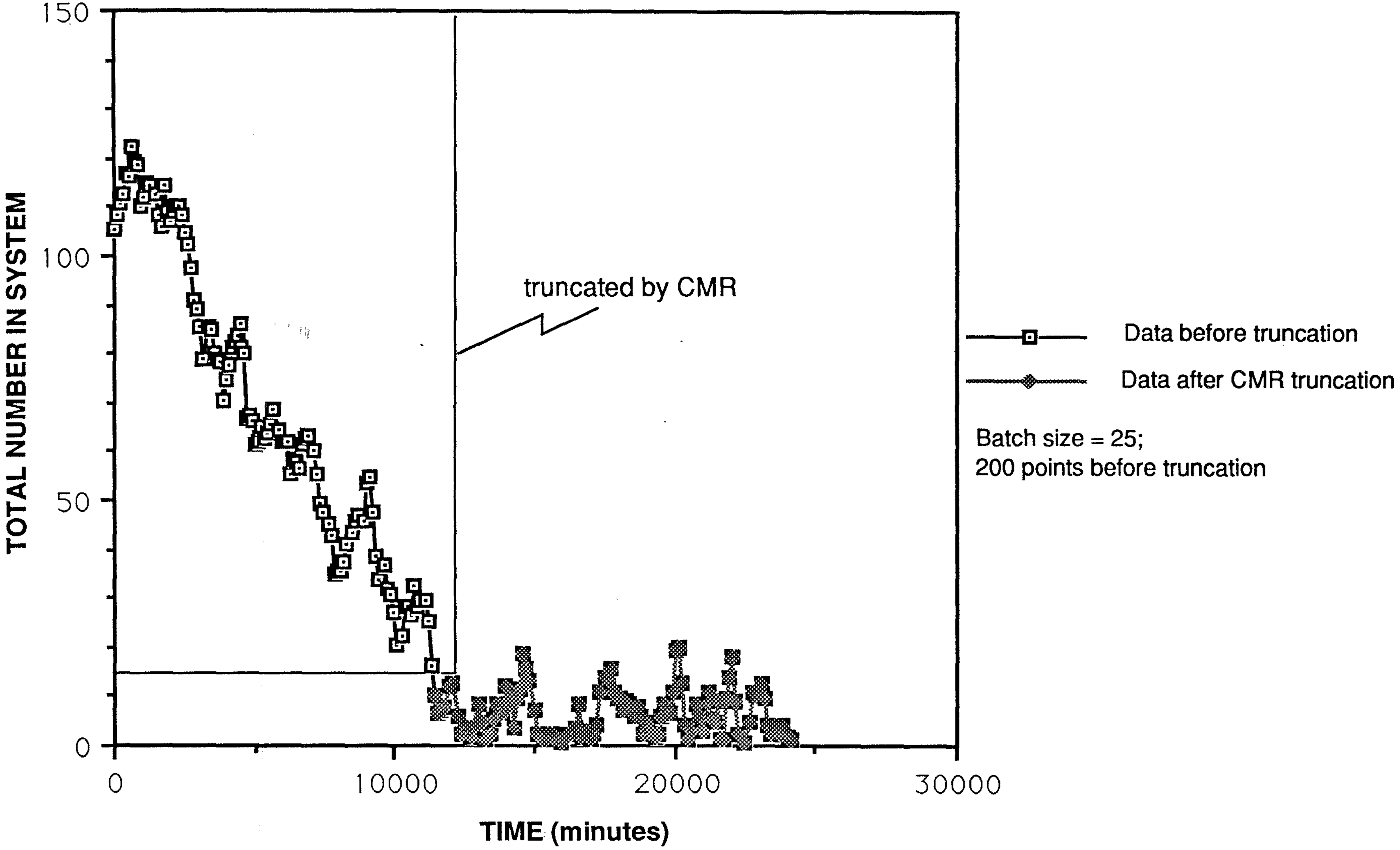

Loaded Queue Model : Run 6
100
80
60
40
20
0
TOTAL NUMBER IN SYSTEM
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

Loaded Queue Model : Run 7



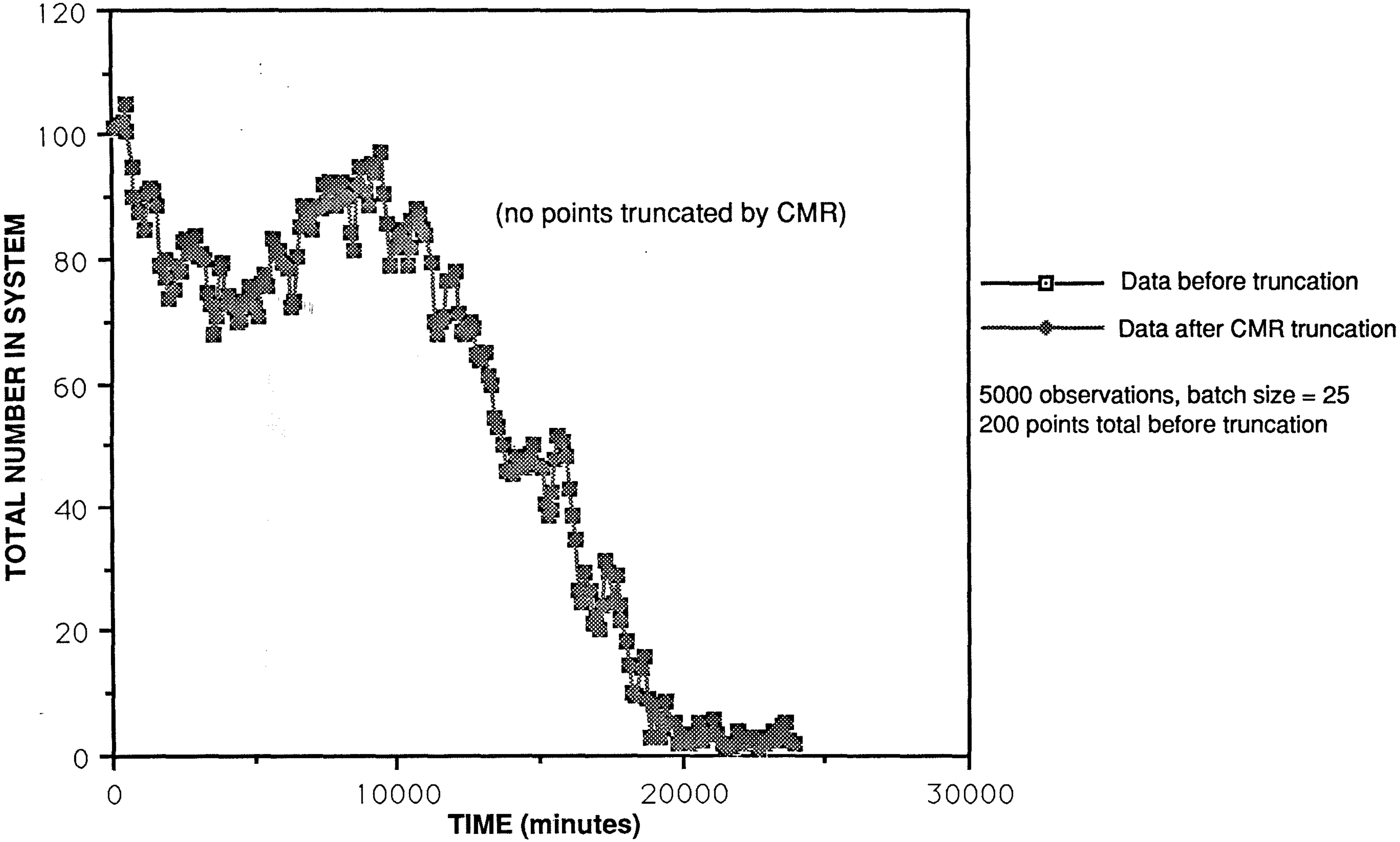
Batch size = 25;
200 points before truncation

Loaded Queue Model : Run 8



Batch size = 25;
200 points before truncation

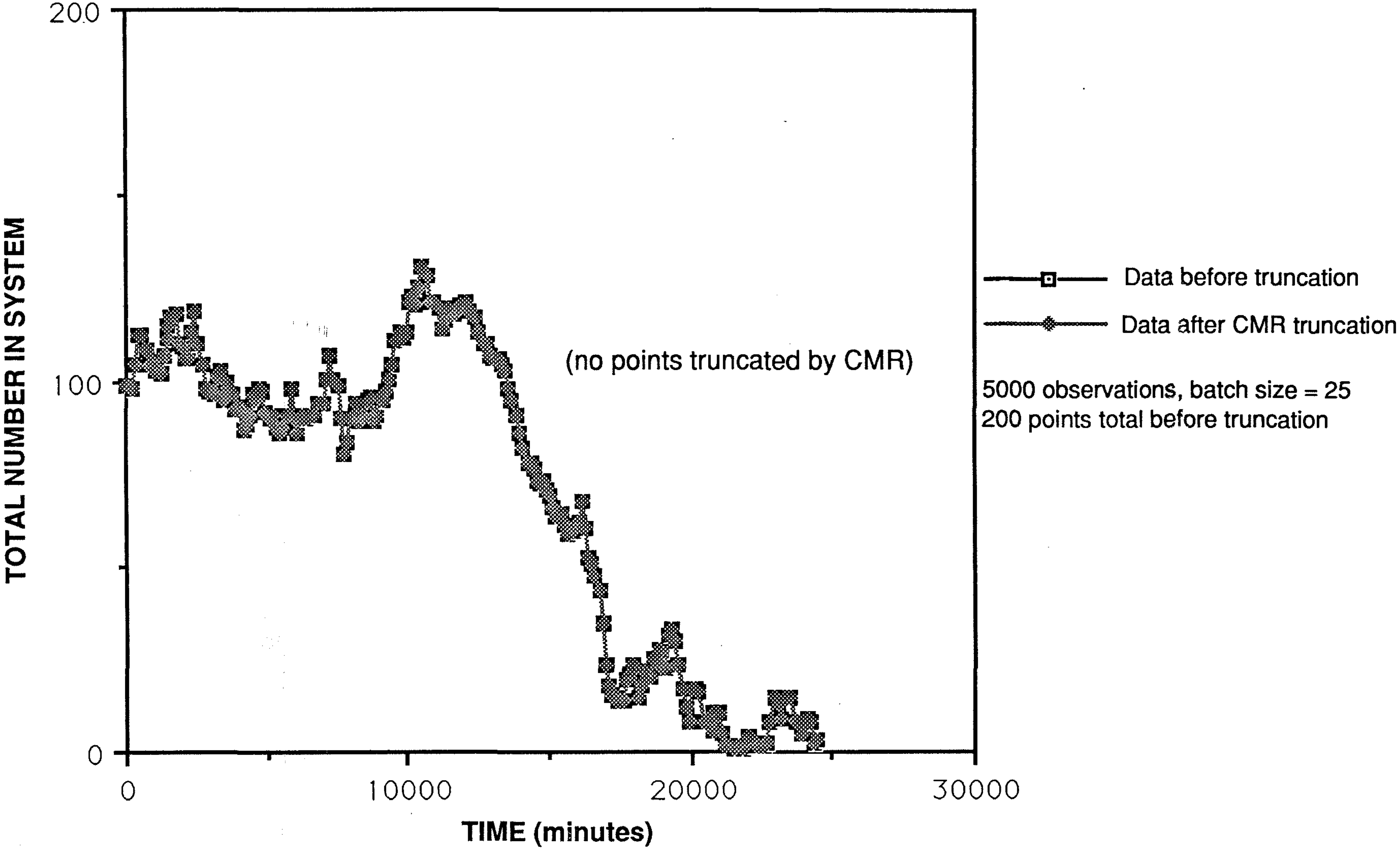
Loaded Queue Model : Run 9



(no points truncated by CMR)

5000 observations, batch size = 25
200 points total before truncation

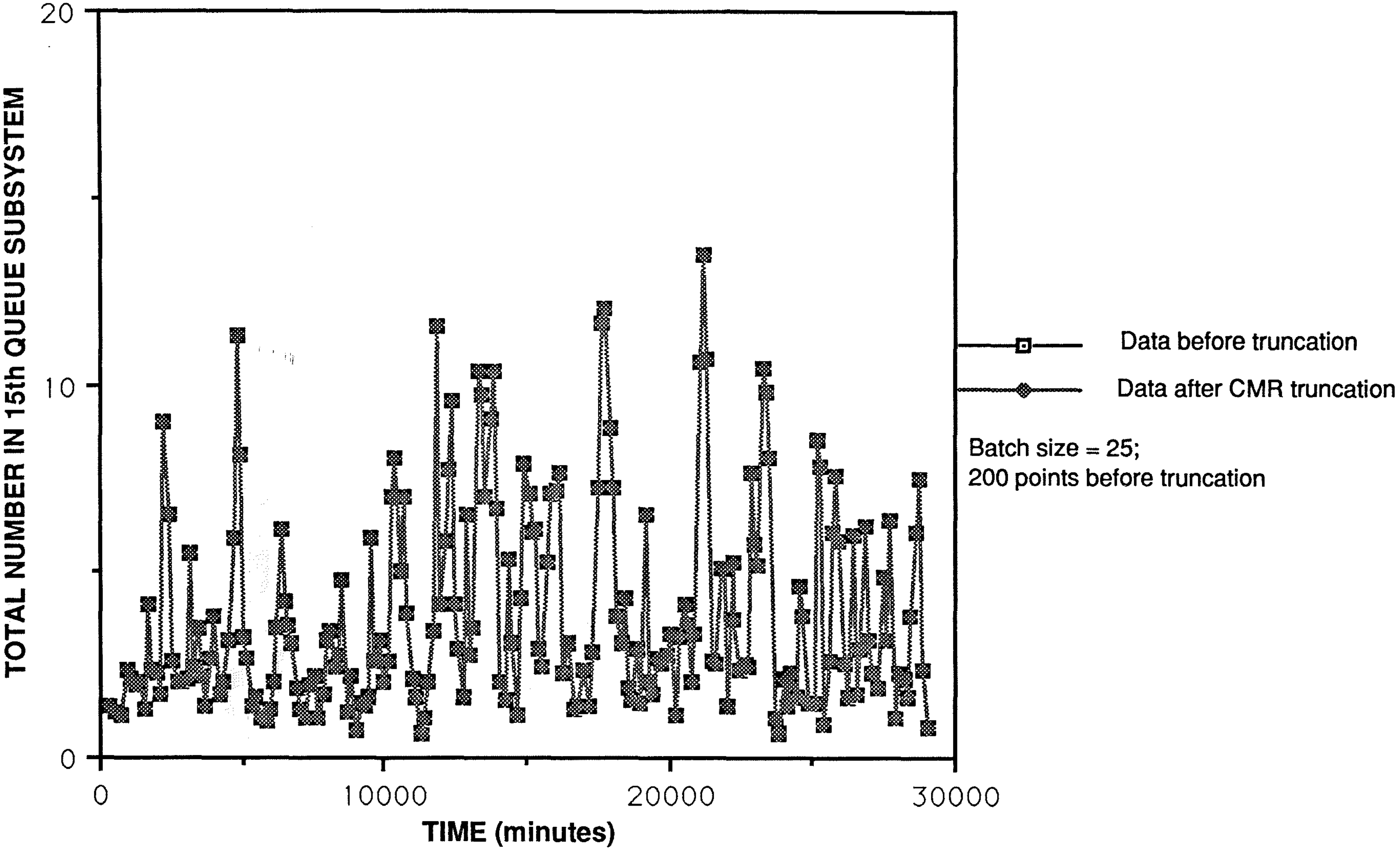
Loaded Queue Model : Run 10



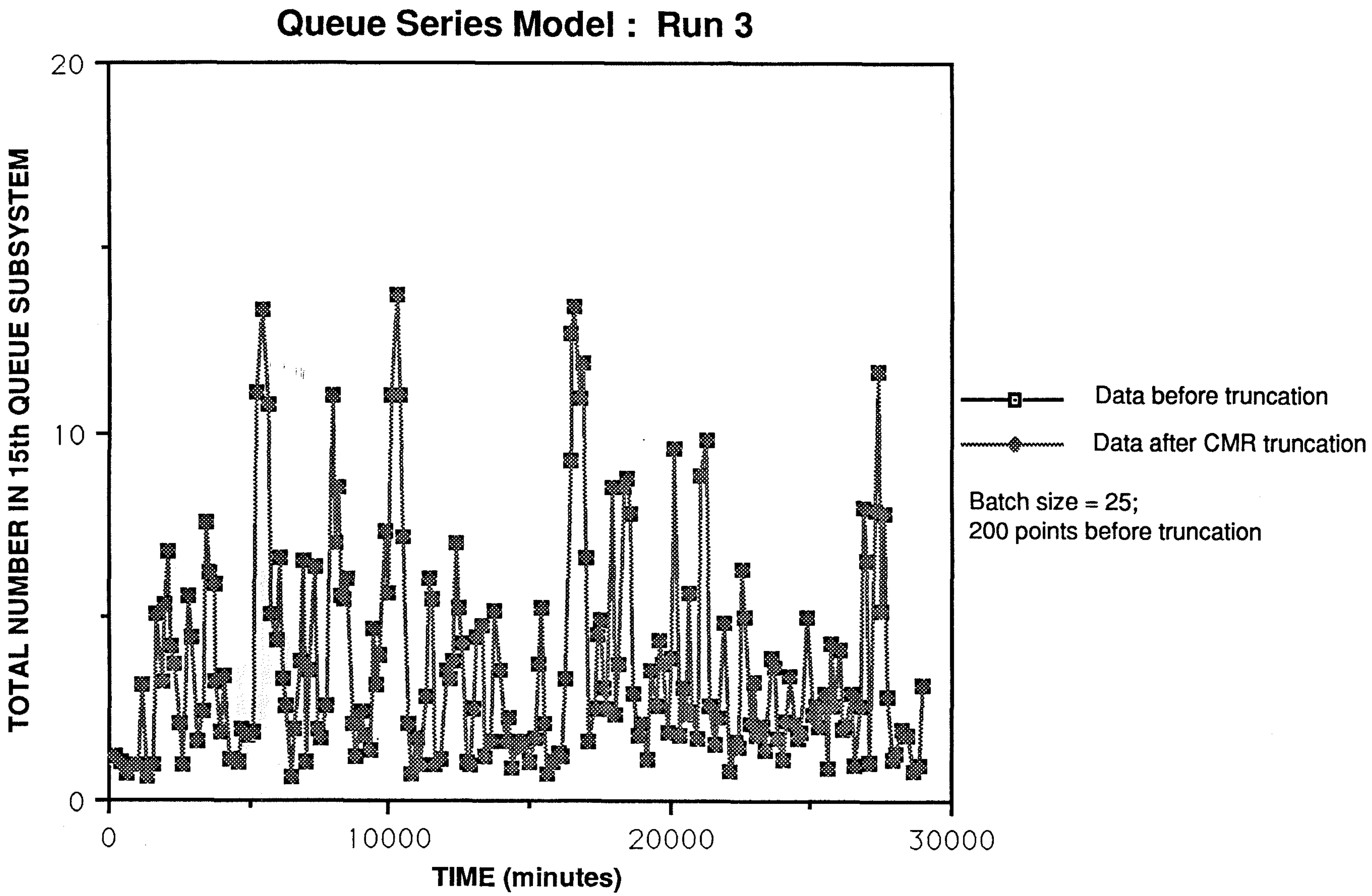
(no points truncated by CMR)

5000 observations, batch size = 25
200 points total before truncation

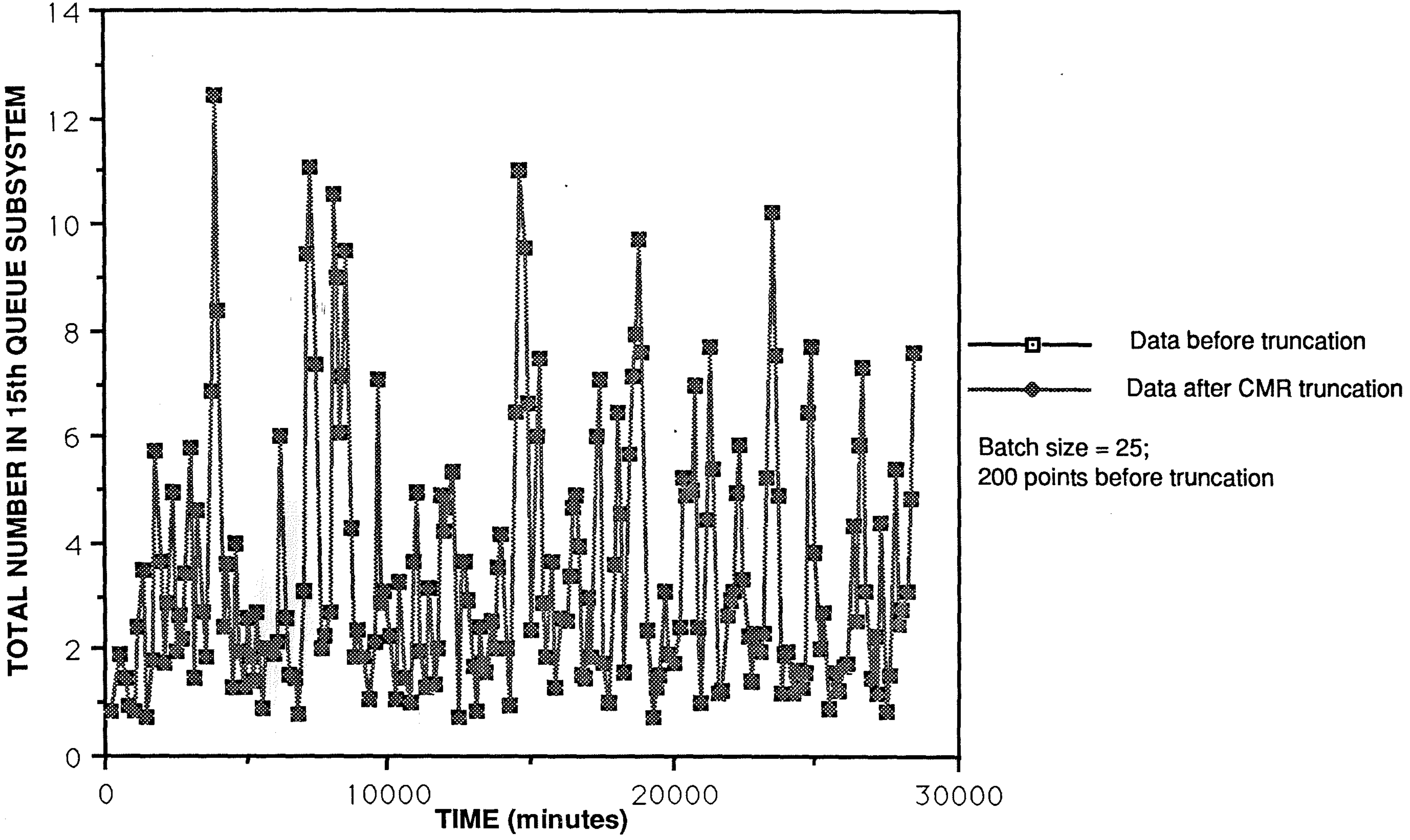
Queue Series Model : Run 2



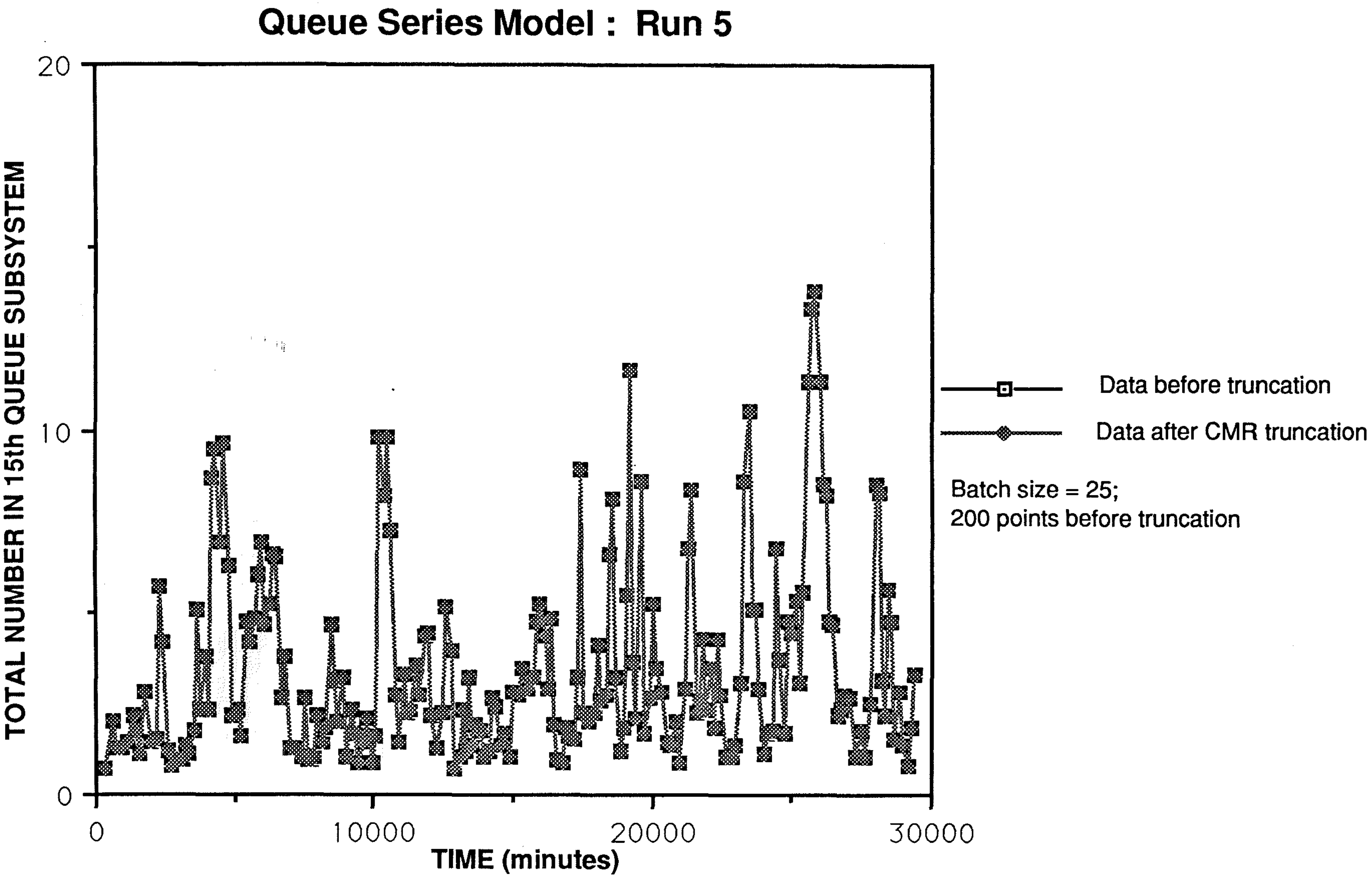
Batch size = 25;
200 points before truncation


Queue Series Model : Run 3
TOTAL NUMBER IN 15th QUEUE SUBSYSTEM
20
10
0
0
10000
20000
30000
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

Queue Series Model : Run 4

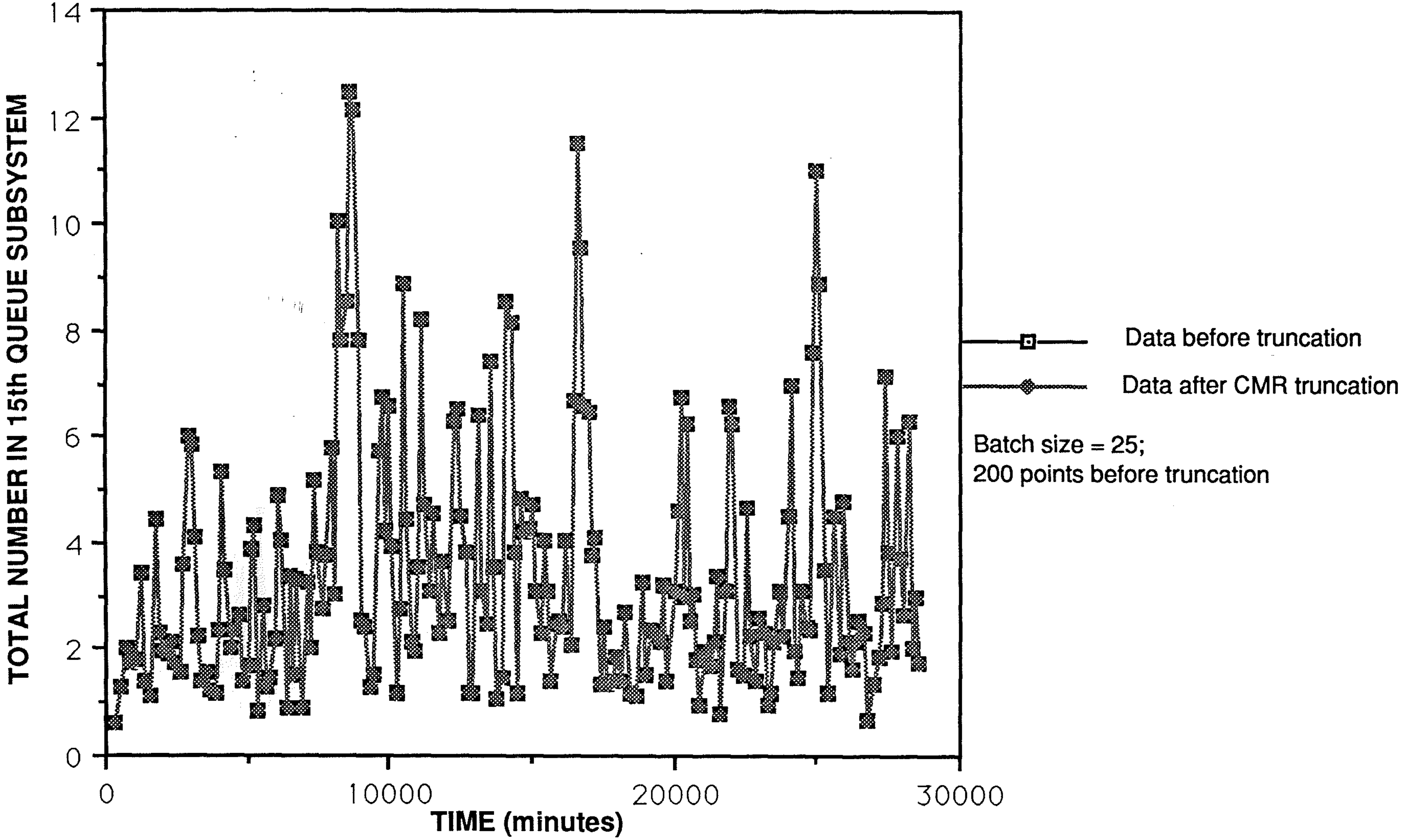


Batch size = 25;
200 points before truncation

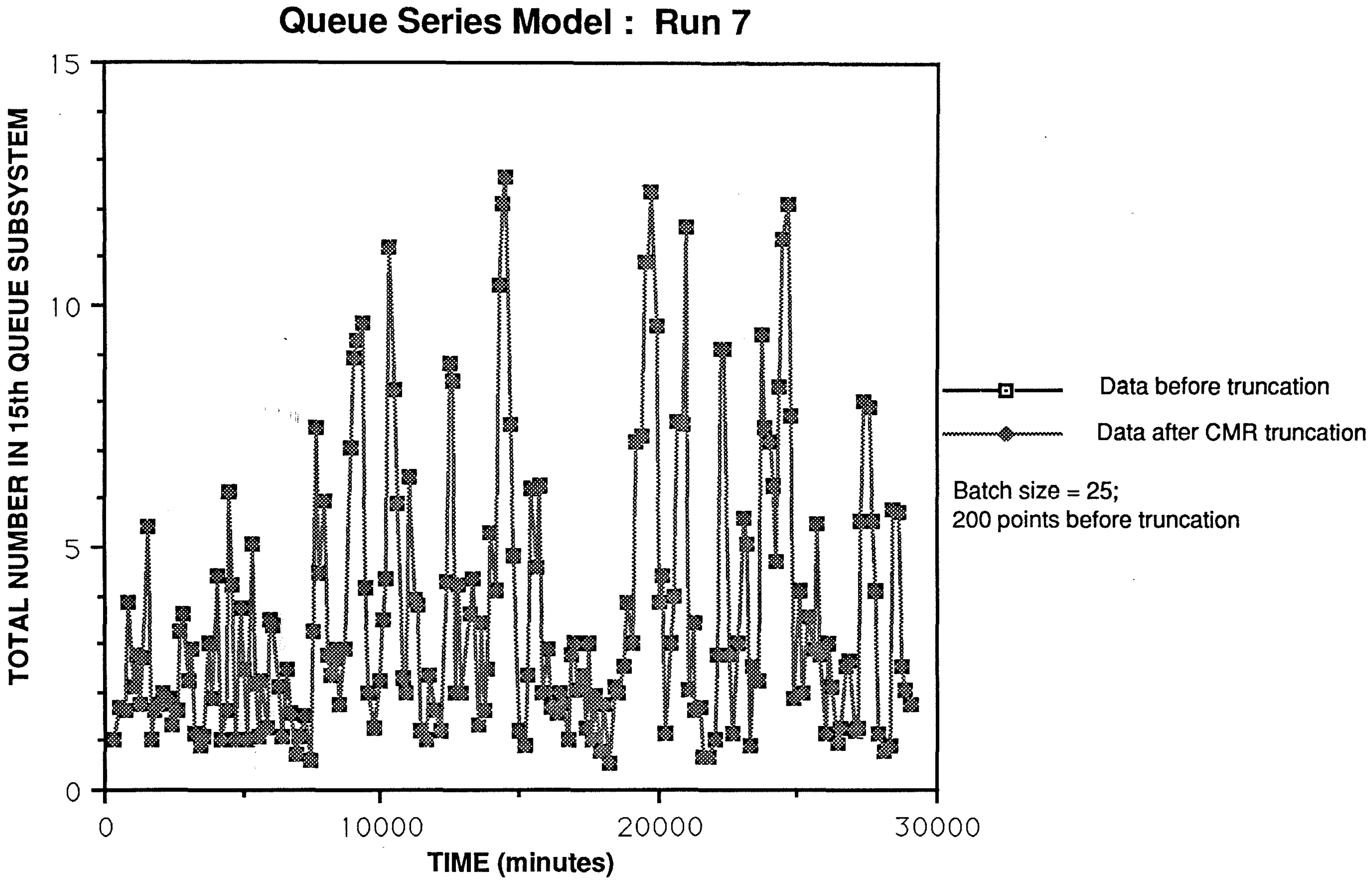


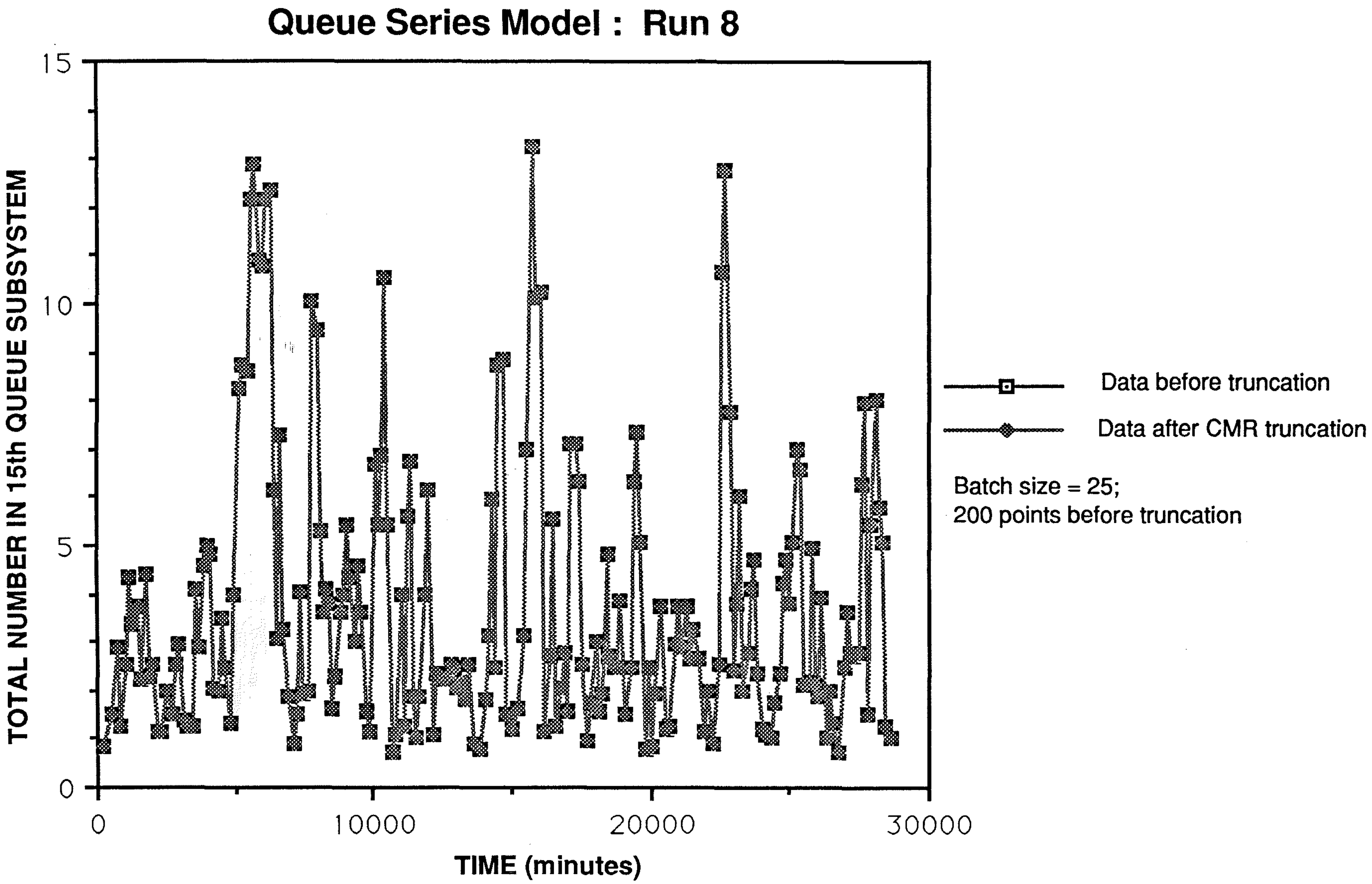
Batch size = 25;
200 points before truncation

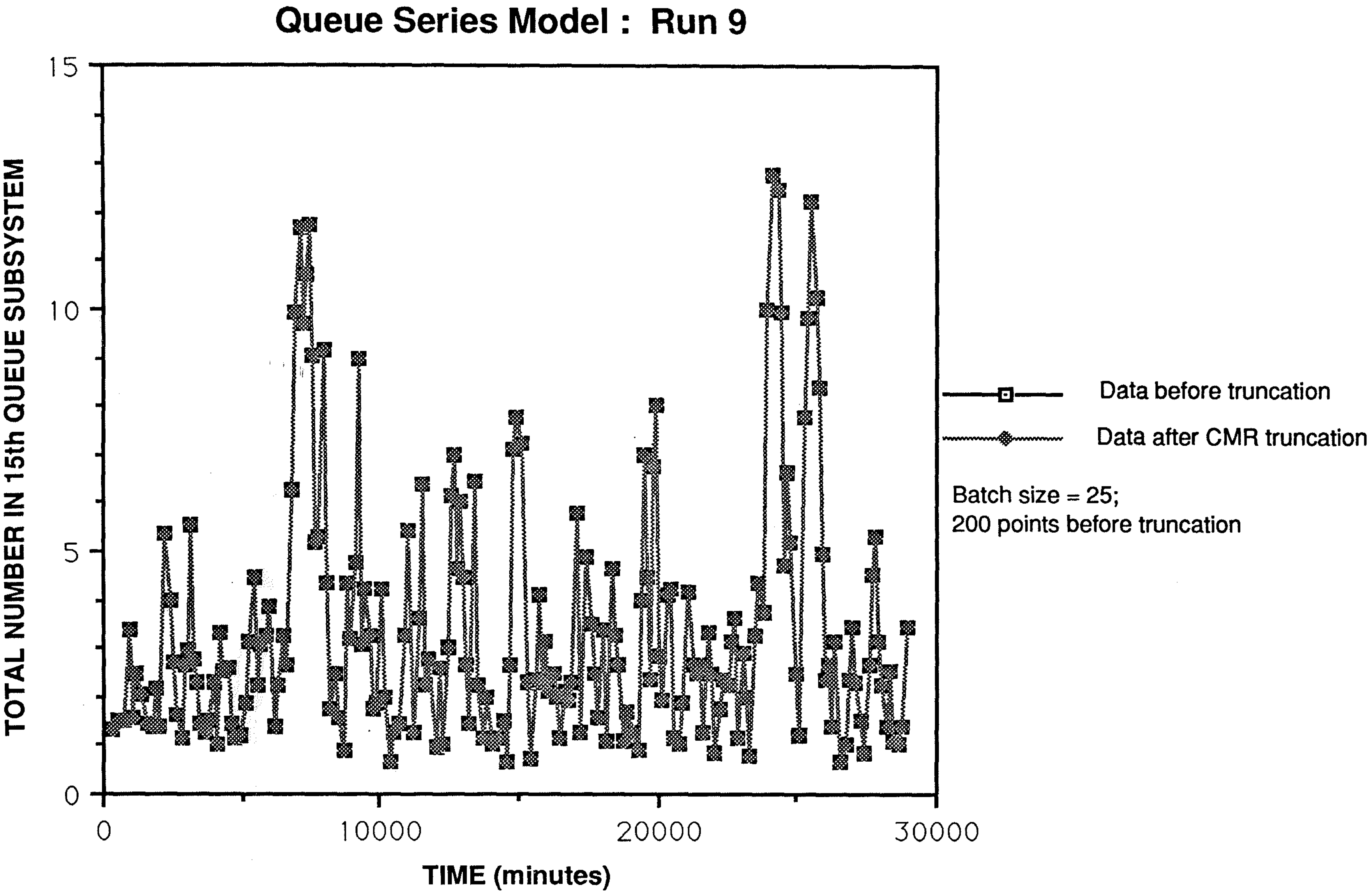
Queue Series Model : Run 6



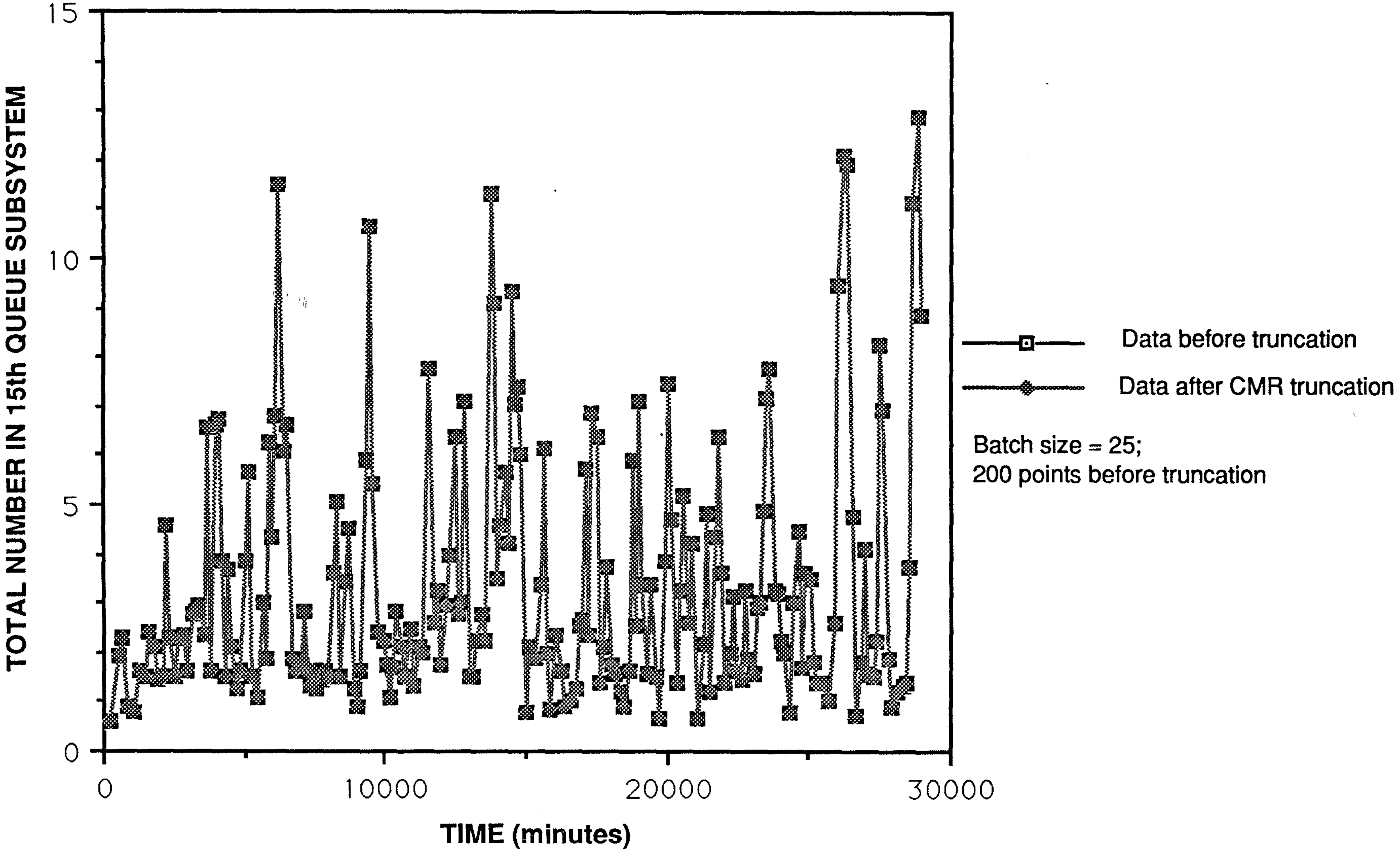
Batch size = 25;
200 points before truncation


Queue Series Model : Run 7
TOTAL NUMBER IN 15th QUEUE SUBSYSTEM
0
5
10
15
TIME (minutes)
0
10000
20000
30000
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

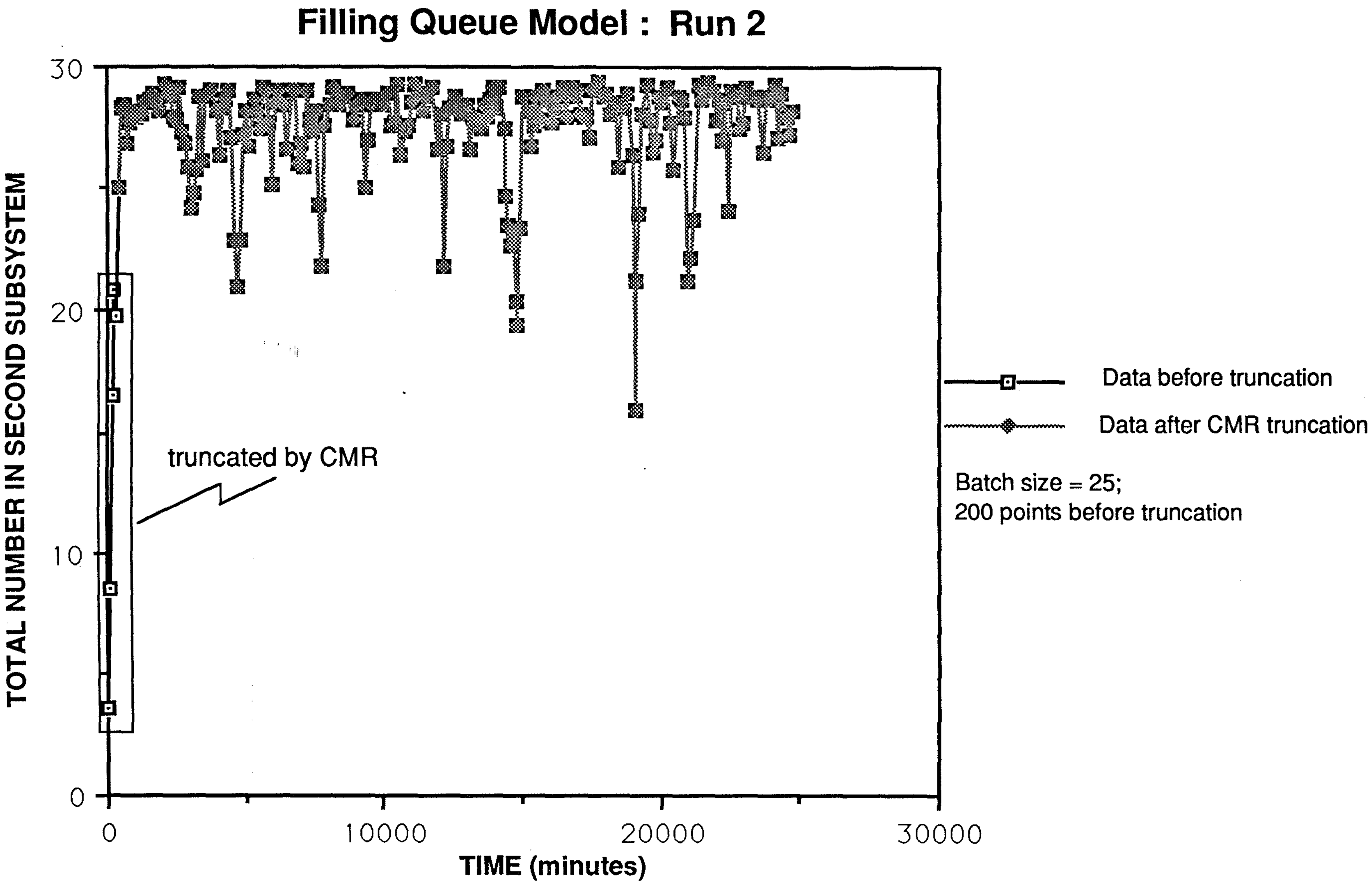

Queue Series Model : Run 8
15
10
5
0
0
10000
20000
30000
TOTAL NUMBER IN 15th QUEUE SUBSYSTEM
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

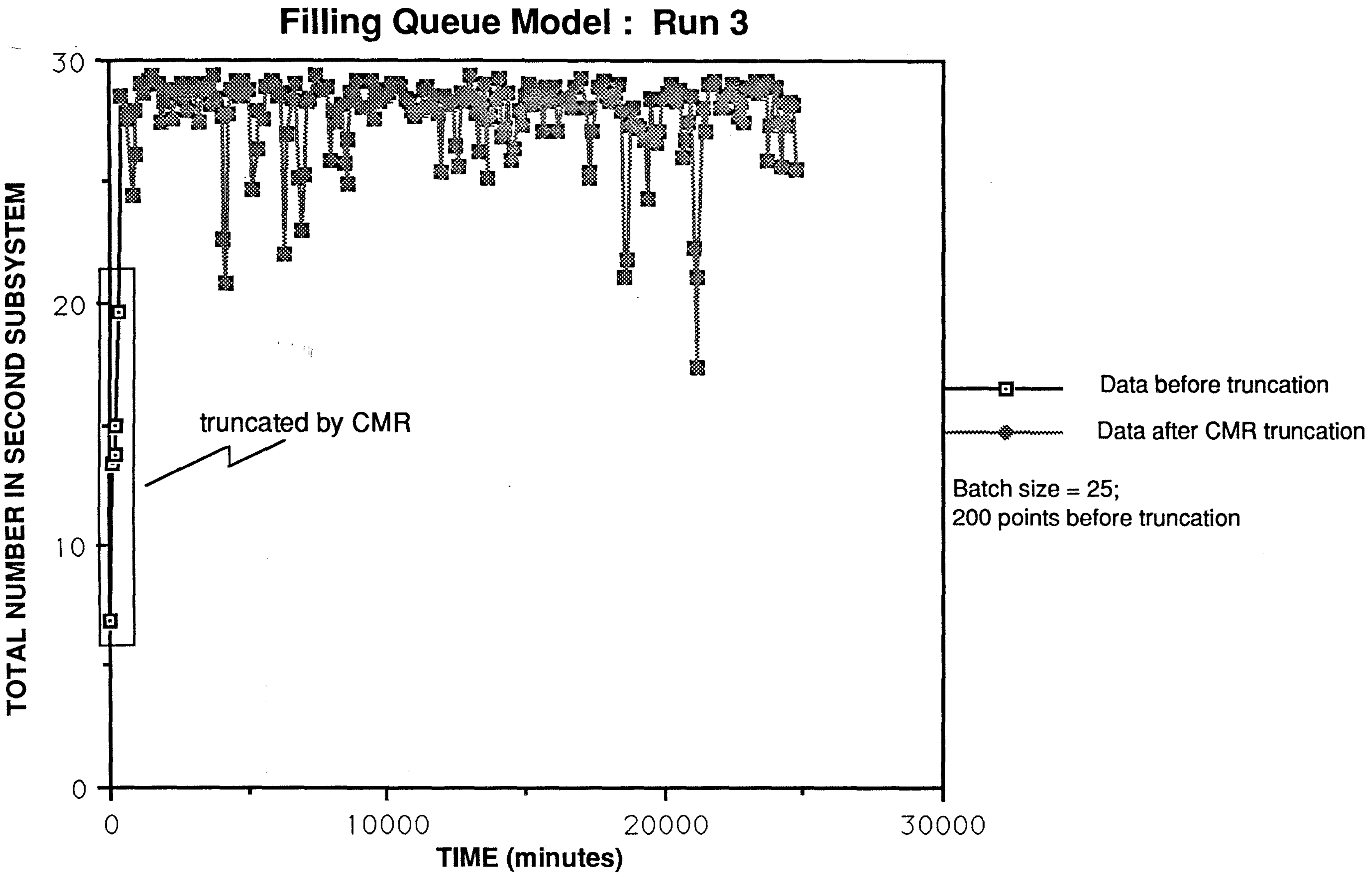

Queue Series Model : Run 9
15
10
5
0
TOTAL NUMBER IN 15th QUEUE SUBSYSTEM
0
10000
20000
30000
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

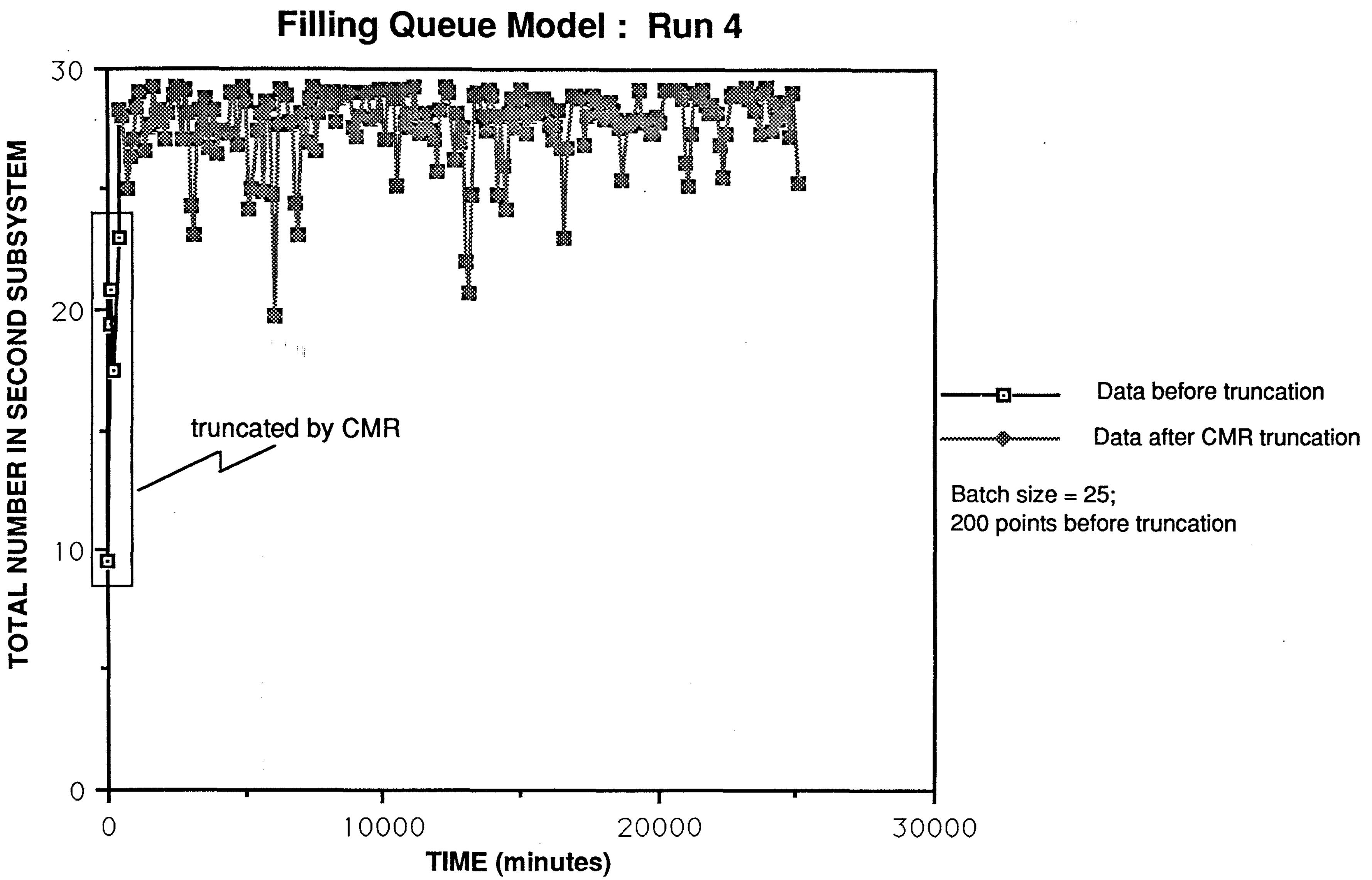
Queue Series Model : Run 10

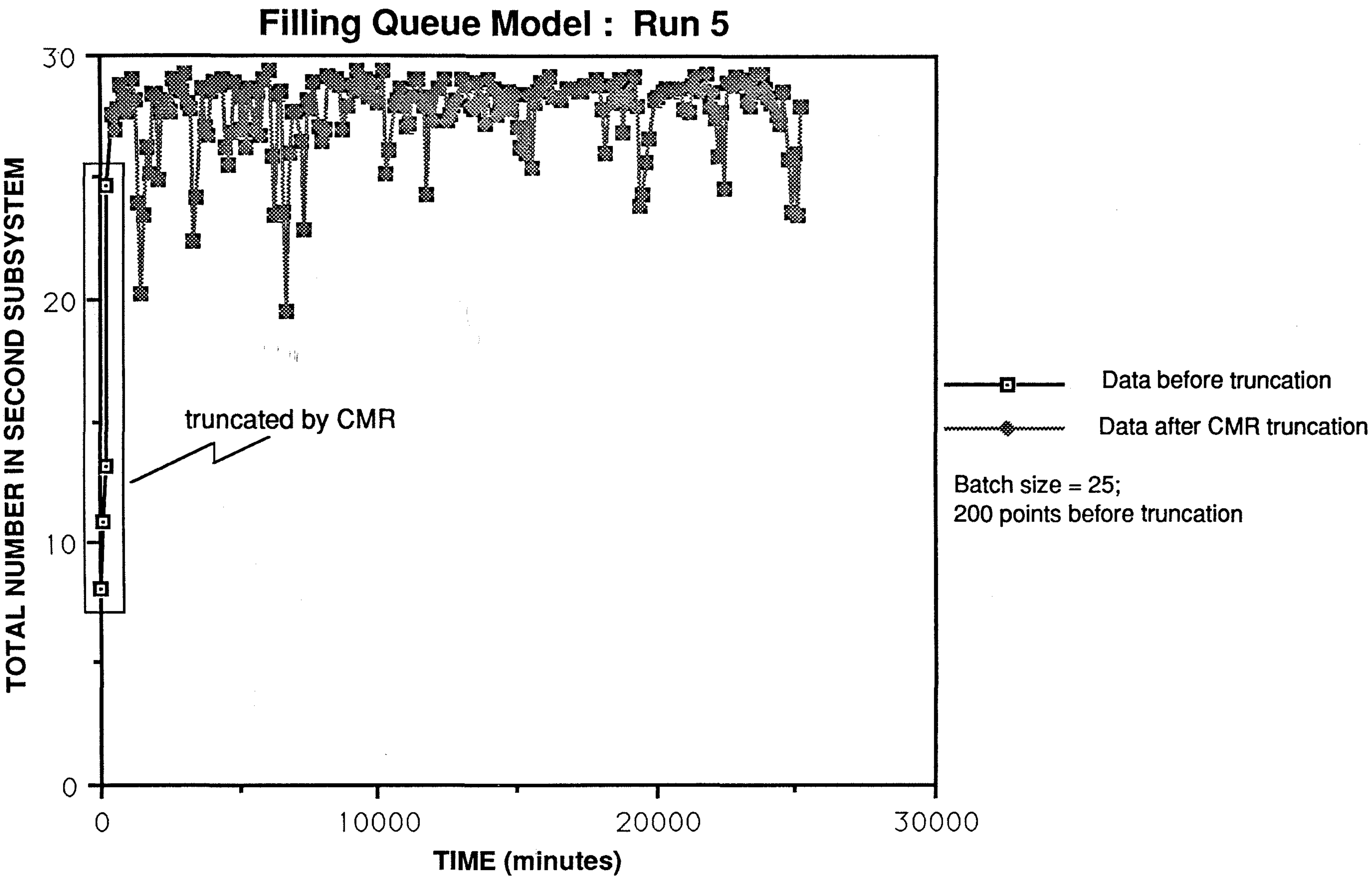


Batch size = 25;
200 points before truncation

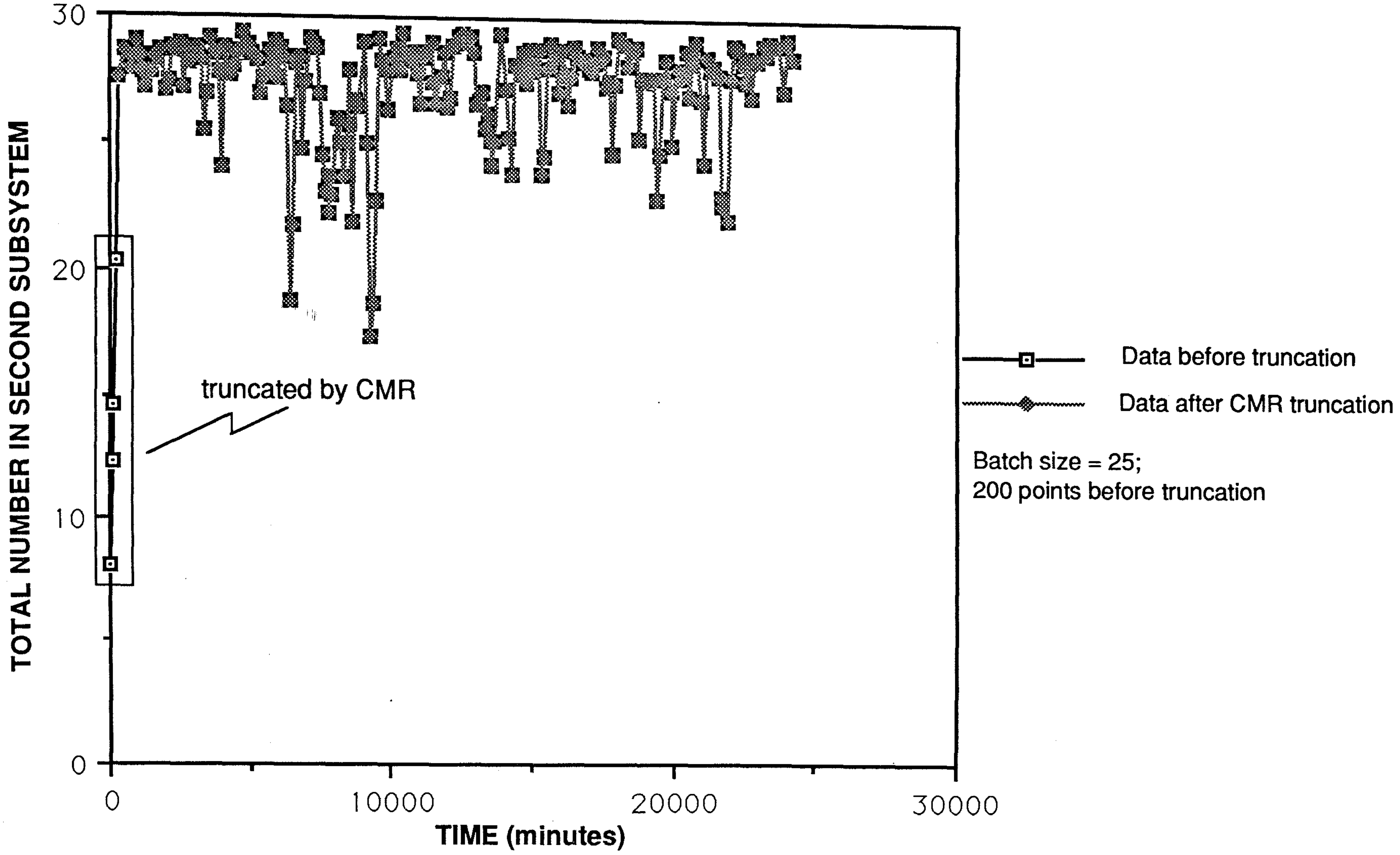

Filling Queue Model : Run 2
TOTAL NUMBER IN SECOND SUBSYSTEM
30
20
10
0
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation


Filling Queue Model : Run 3
TOTAL NUMBER IN SECOND SUBSYSTEM
30
20
10
0
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation


Filling Queue Model : Run 4
30
20
10
0
TOTAL NUMBER IN SECOND SUBSYSTEM
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

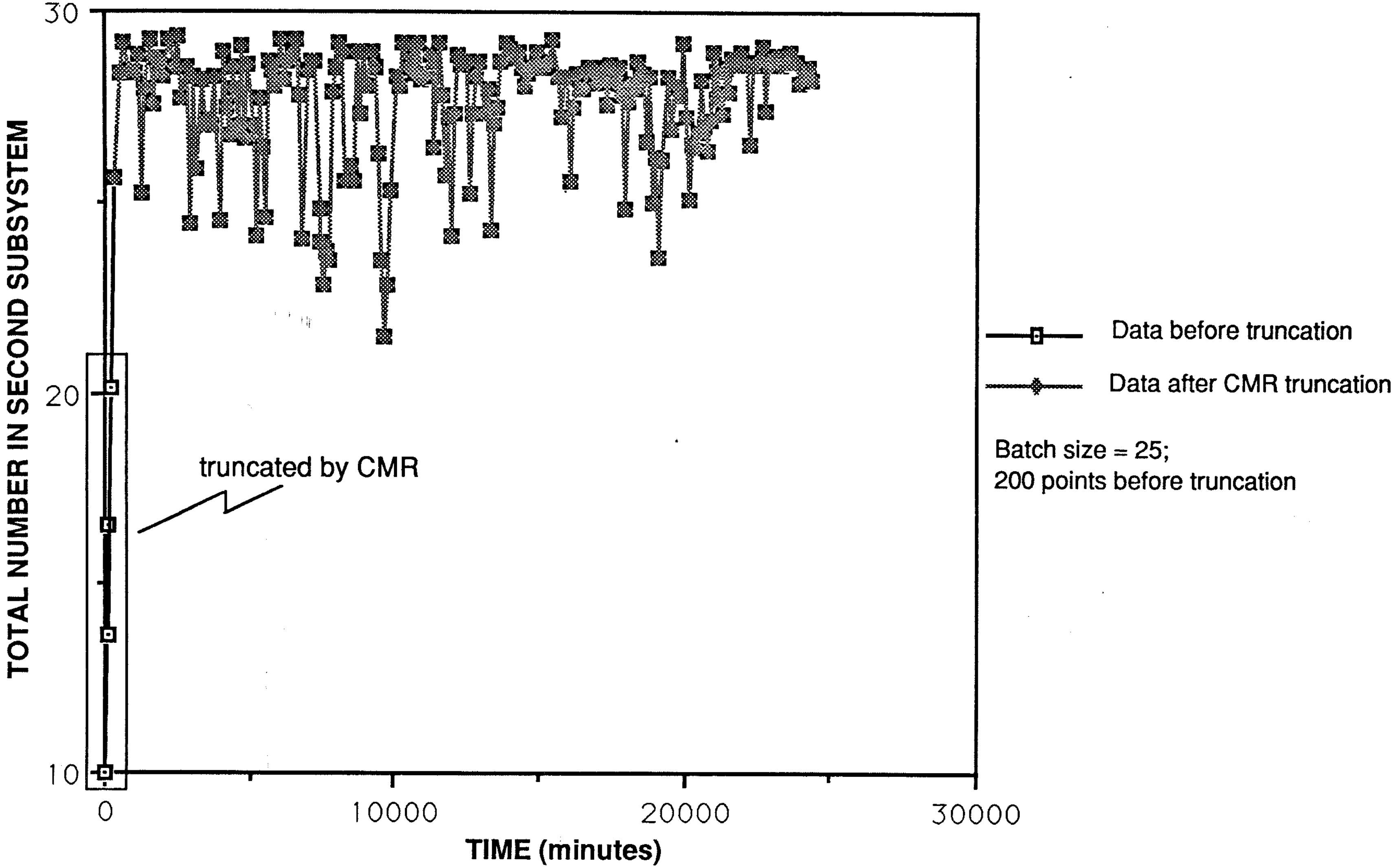

Filling Queue Model : Run 5
TOTAL NUMBER IN SECOND SUBSYSTEM
0
10
20
30
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

Filling Queue Model : Run 6



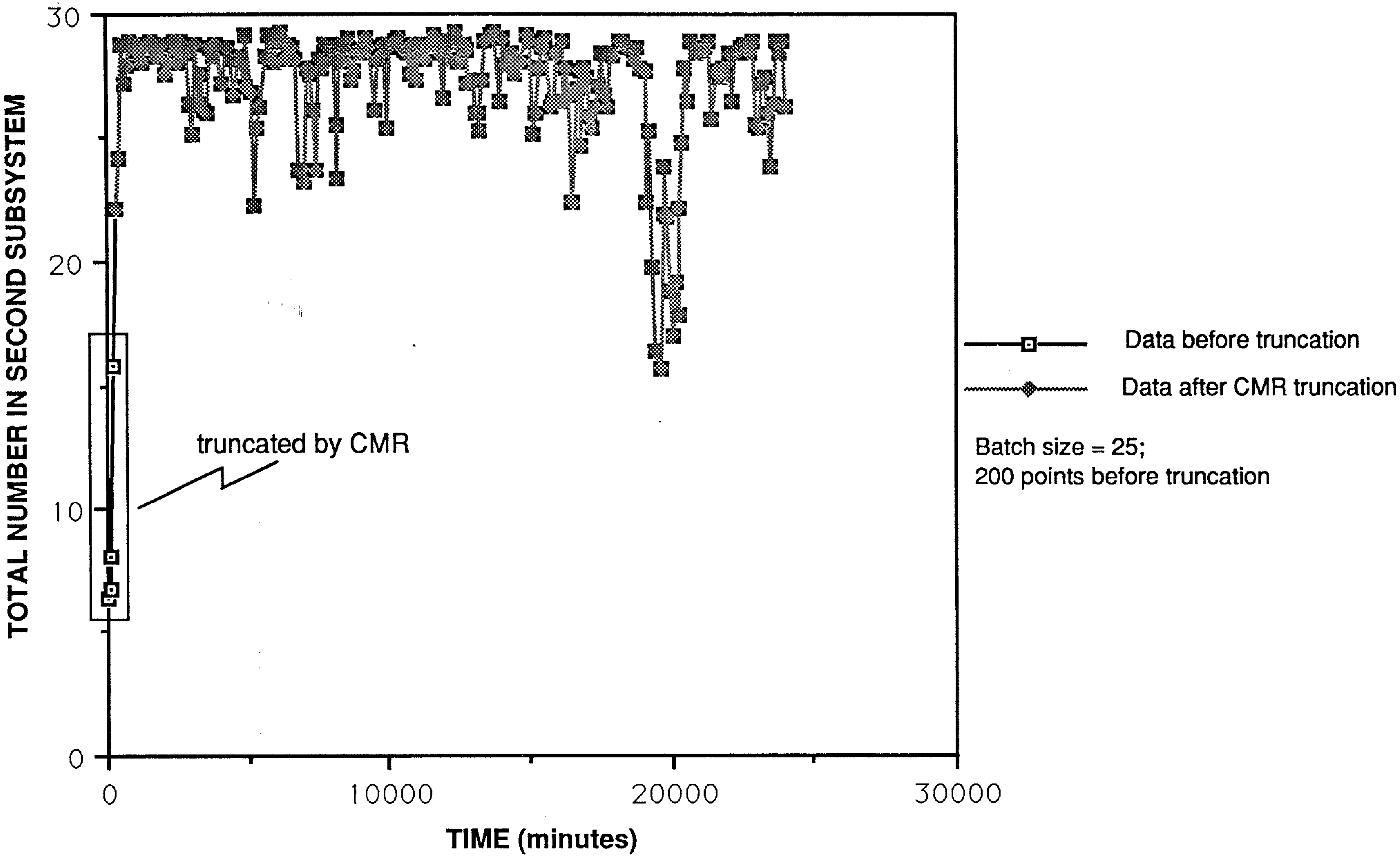
Batch size = 25;
200 points before truncation

Filling Queue Model : Run 7

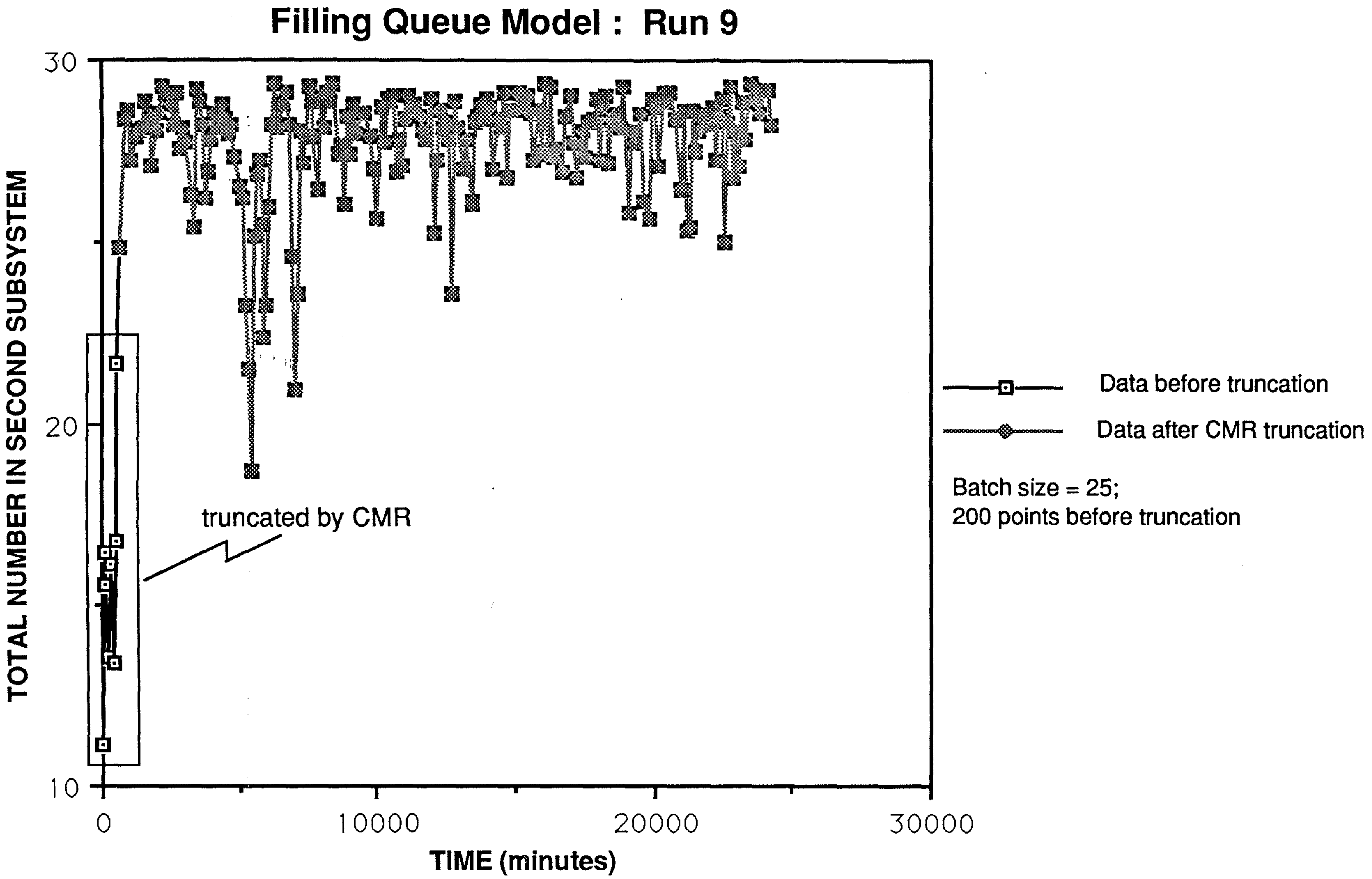


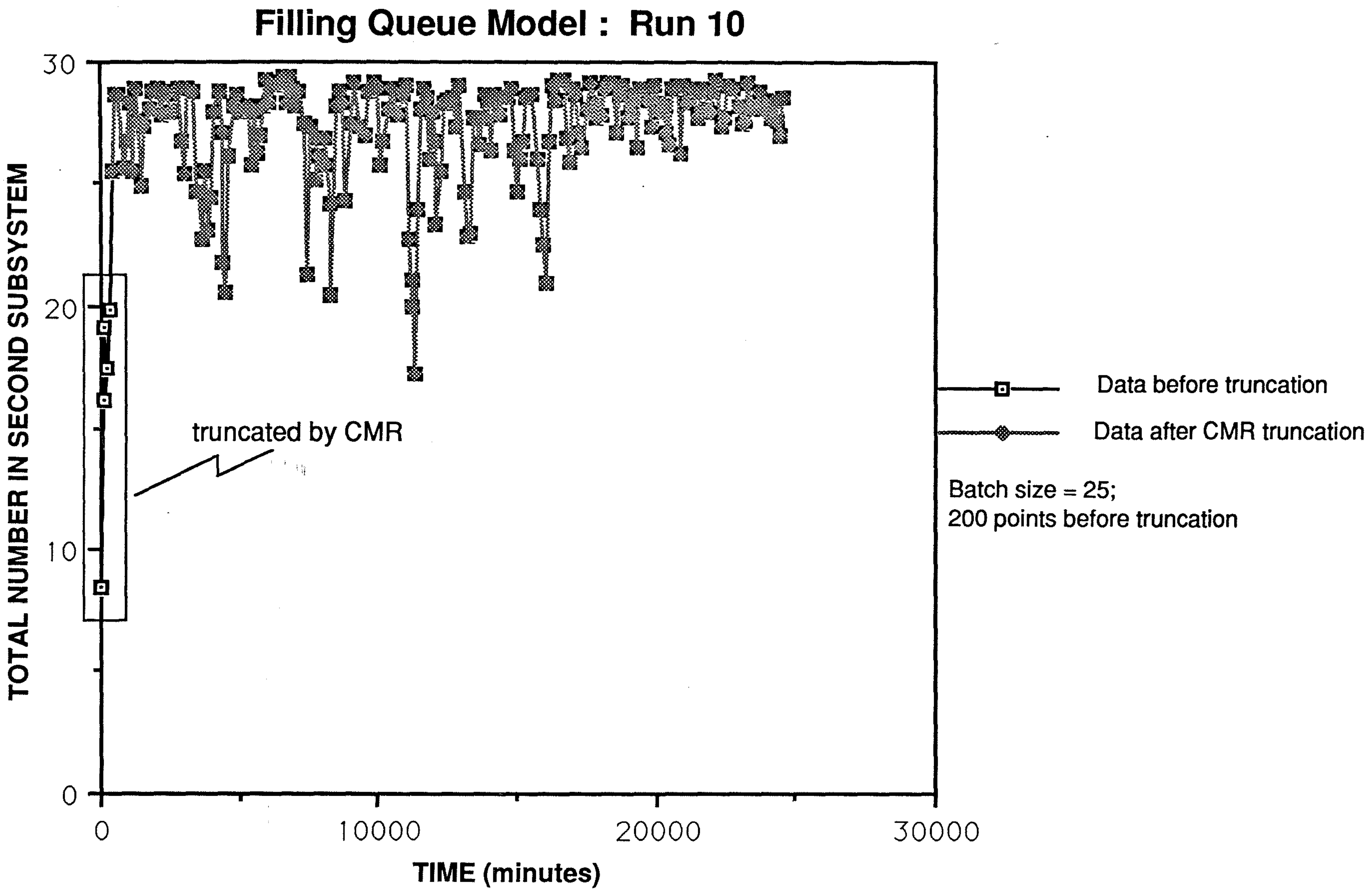
Batch size = 25;
200 points before truncation

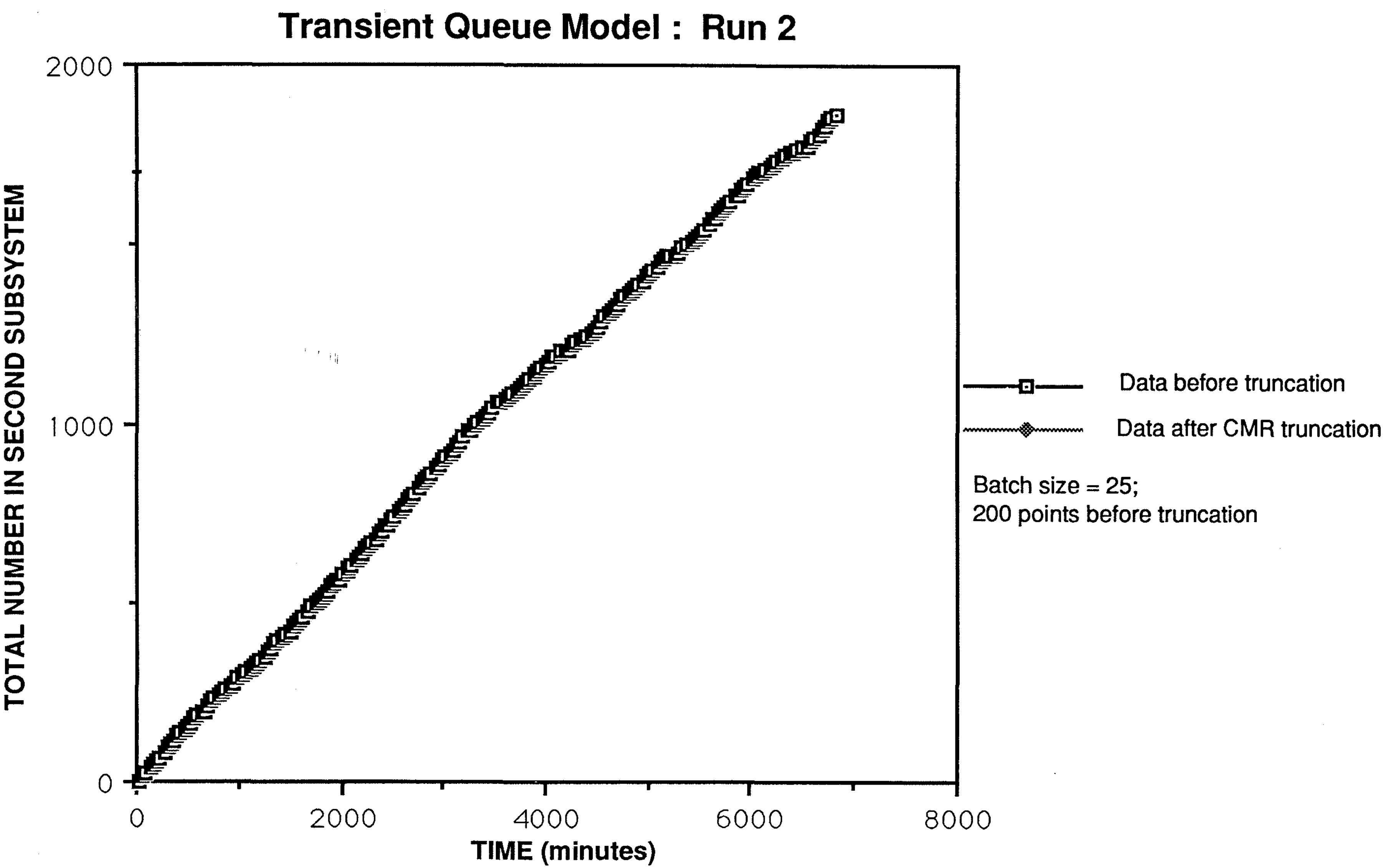
Filling Queue Model : Run 8



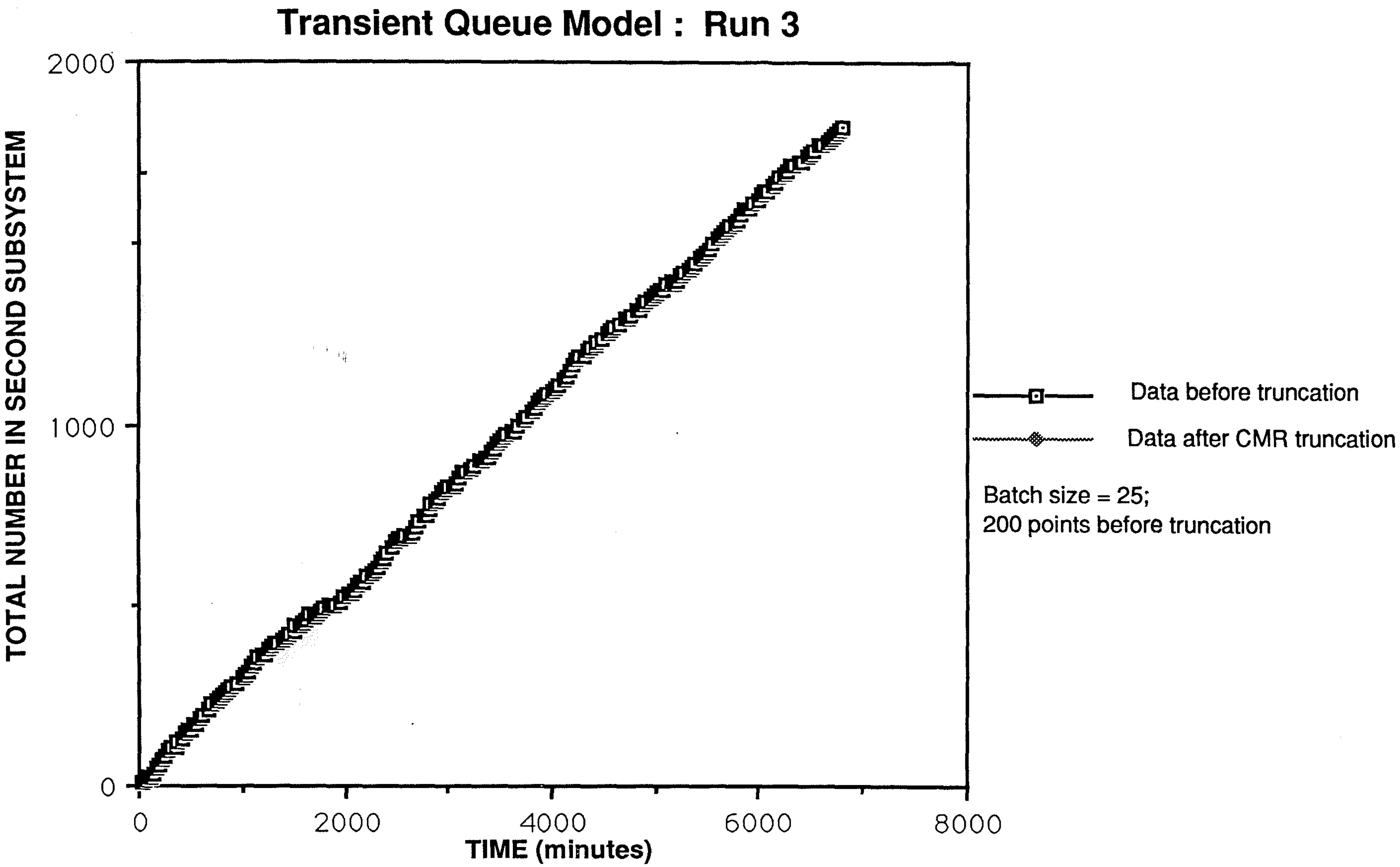
Batch size = 25;
200 points before truncation


Filling Queue Model : Run 9
TOTAL NUMBER IN SECOND SUBSYSTEM
30
20
10
0
10000
20000
30000
TIME (minutes)
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

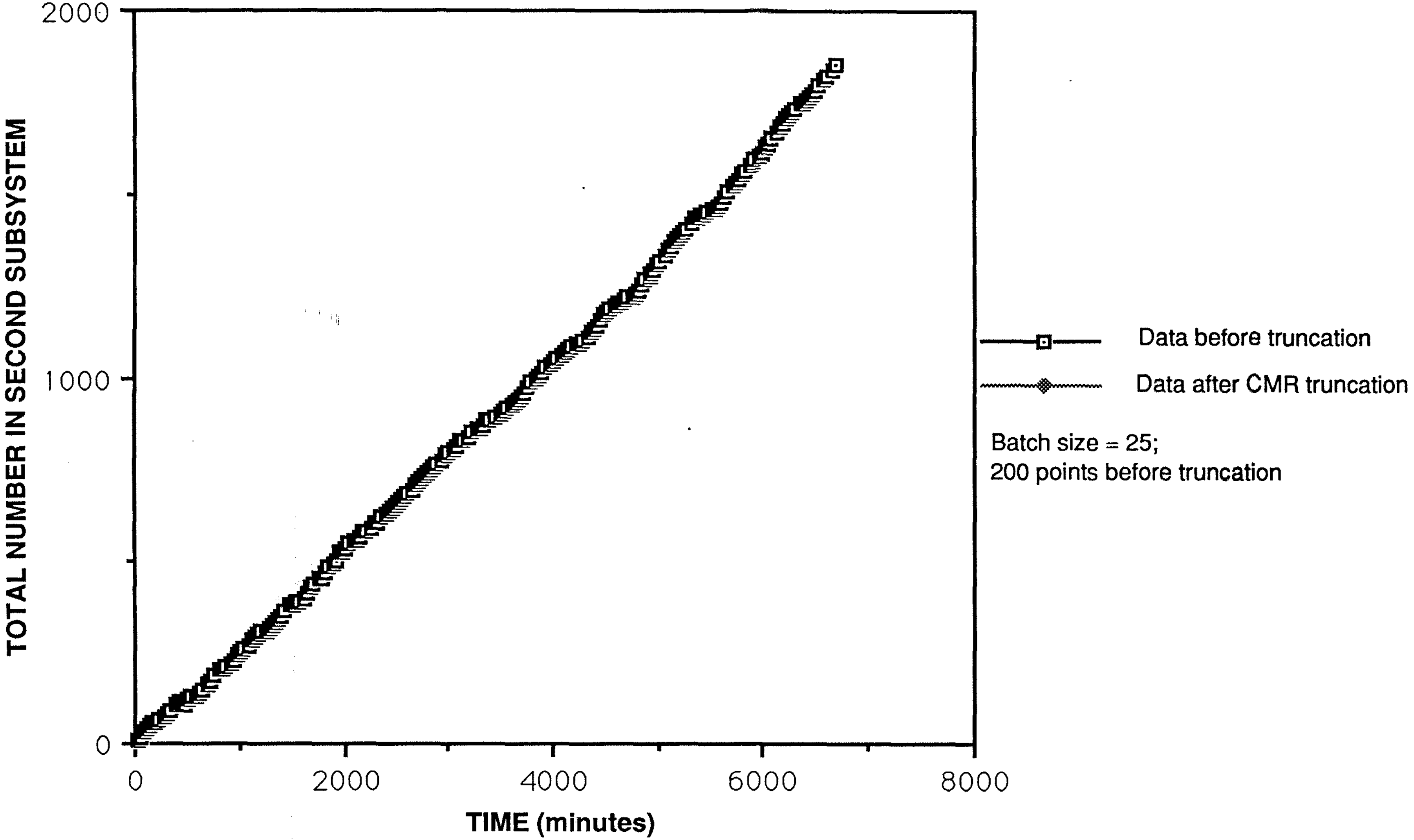

Filling Queue Model : Run 10
30
20
10
0
0
10000
20000
30000
TIME (minutes)
TOTAL NUMBER IN SECOND SUBSYSTEM
truncated by CMR
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation



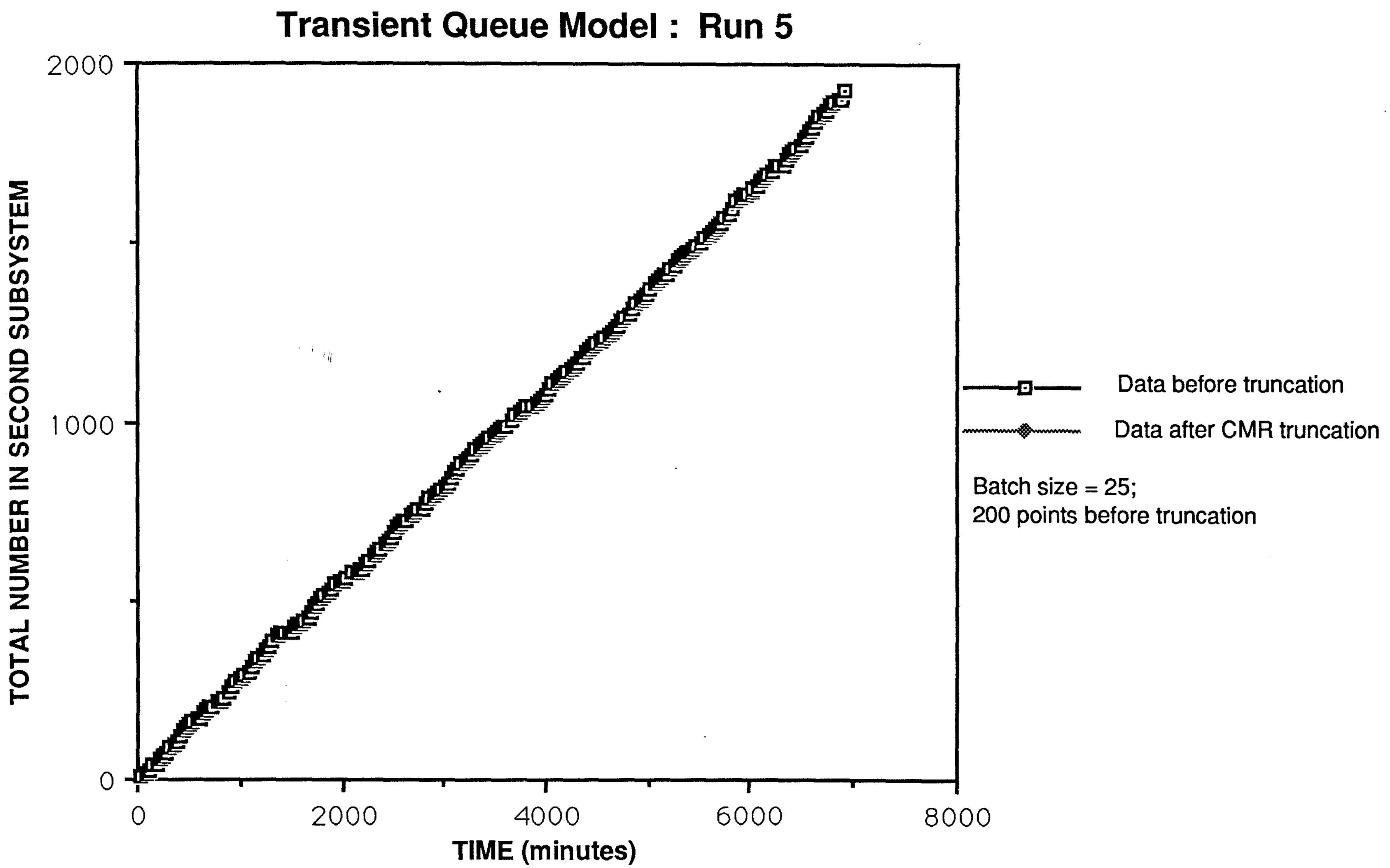
Transient Queue Model : Run 2
2000
1000
0
0
2000
4000
6000
8000
TIME (minutes)
TOTAL NUMBER IN SECOND SUBSYSTEM
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation


Transient Queue Model : Run 3
2000
1000
0
TOTAL NUMBER IN SECOND SUBSYSTEM
0
2000
4000
6000
8000
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

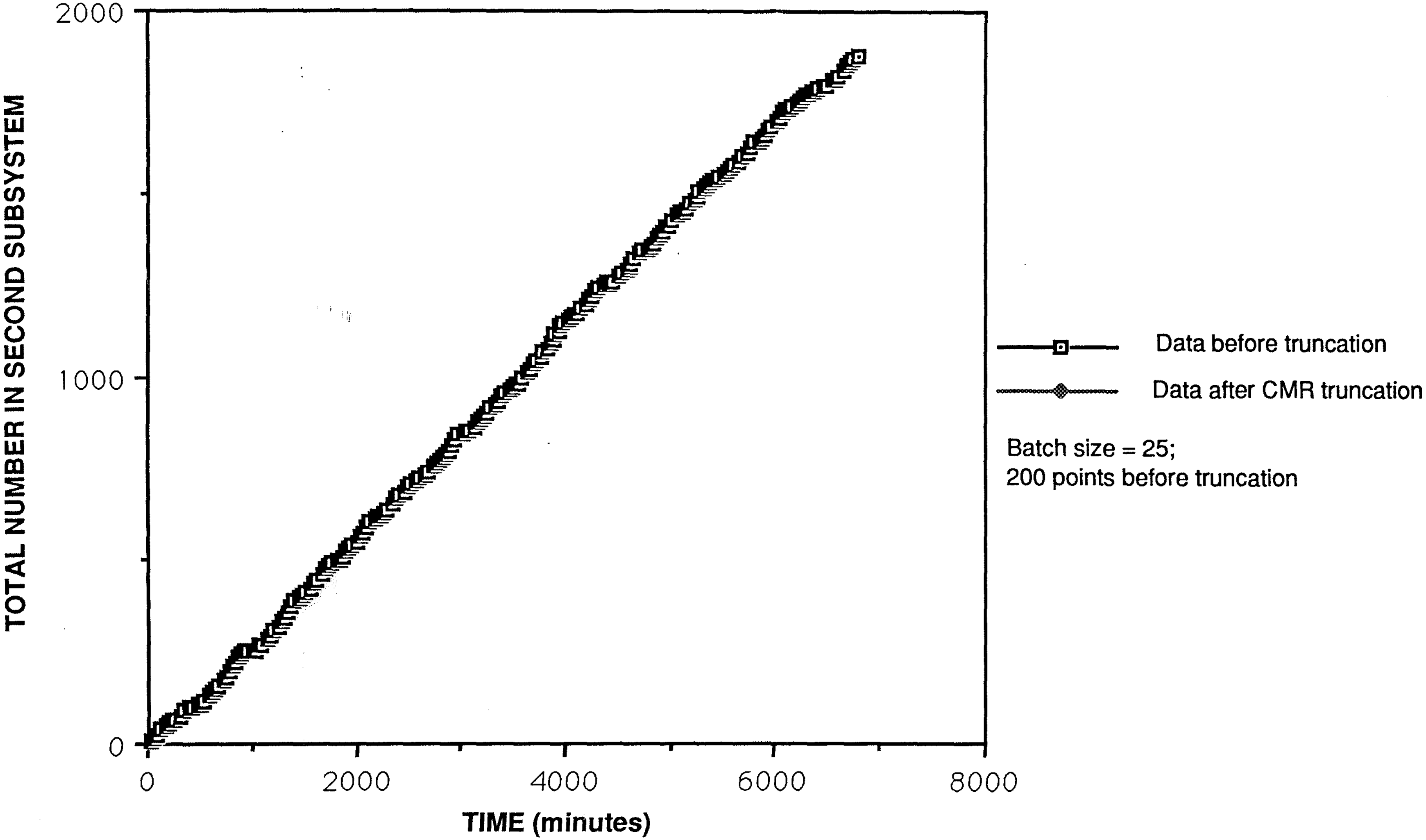
Transient Queue Model : Run 4



Batch size = 25;
200 points before truncation

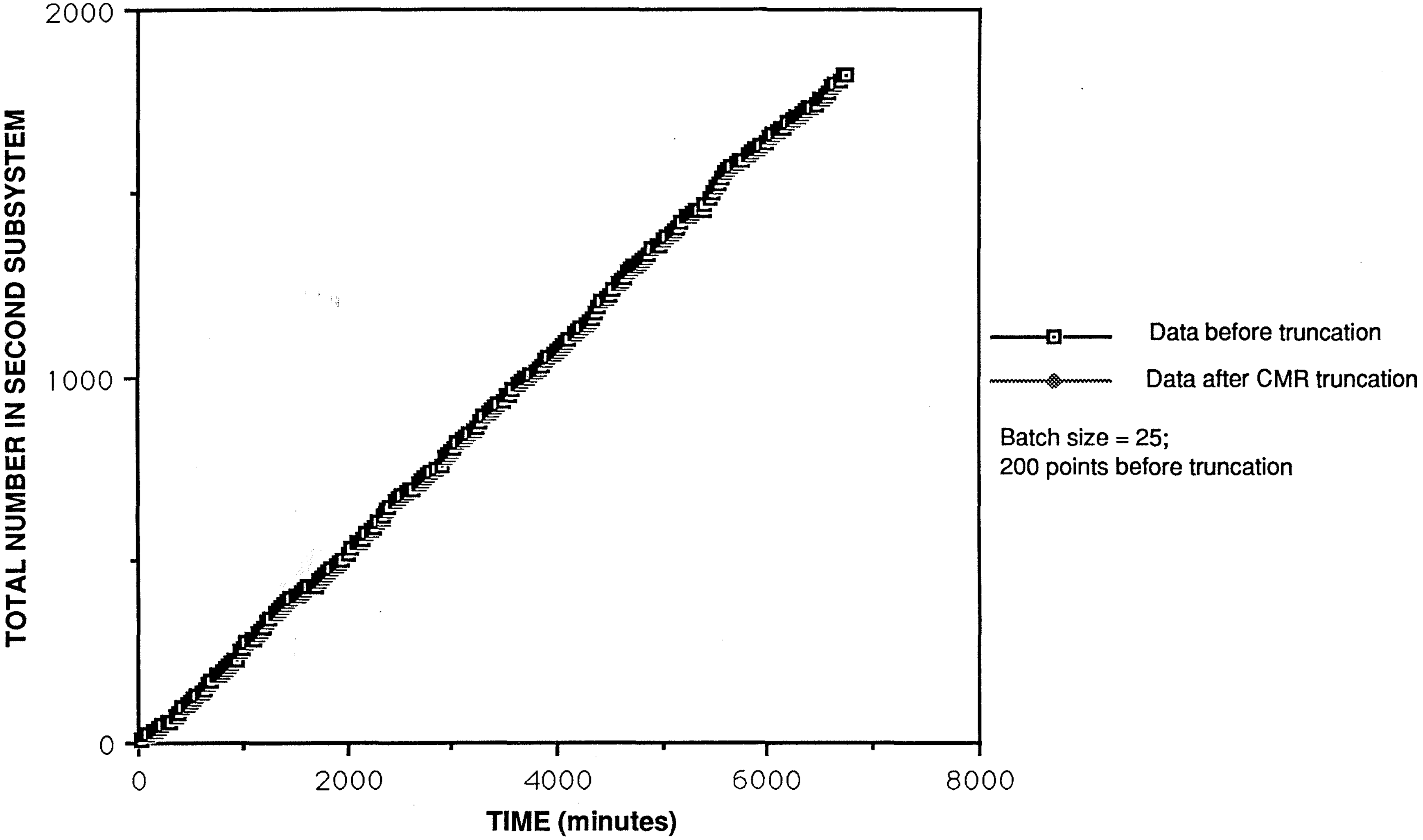

Transient Queue Model : Run 5
2000
1000
0
TOTAL NUMBER IN SECOND SUBSYSTEM
0
2000
4000
6000
8000
TIME (minutes)
Data before truncation
Data after CMR truncation
Batch size = 25;
200 points before truncation

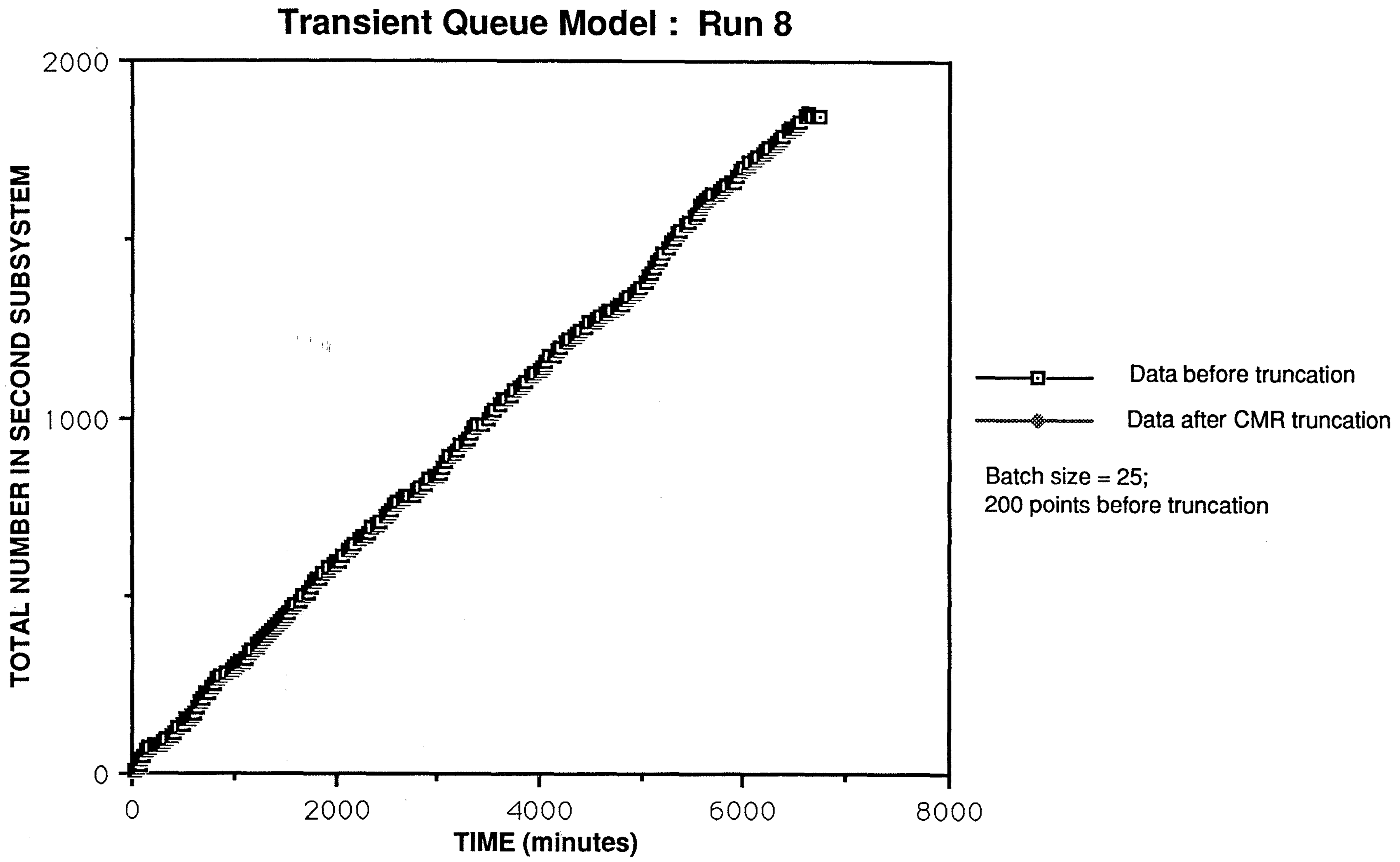
Transient Queue Model : Run 6



Batch size = 25;
200 points before truncation

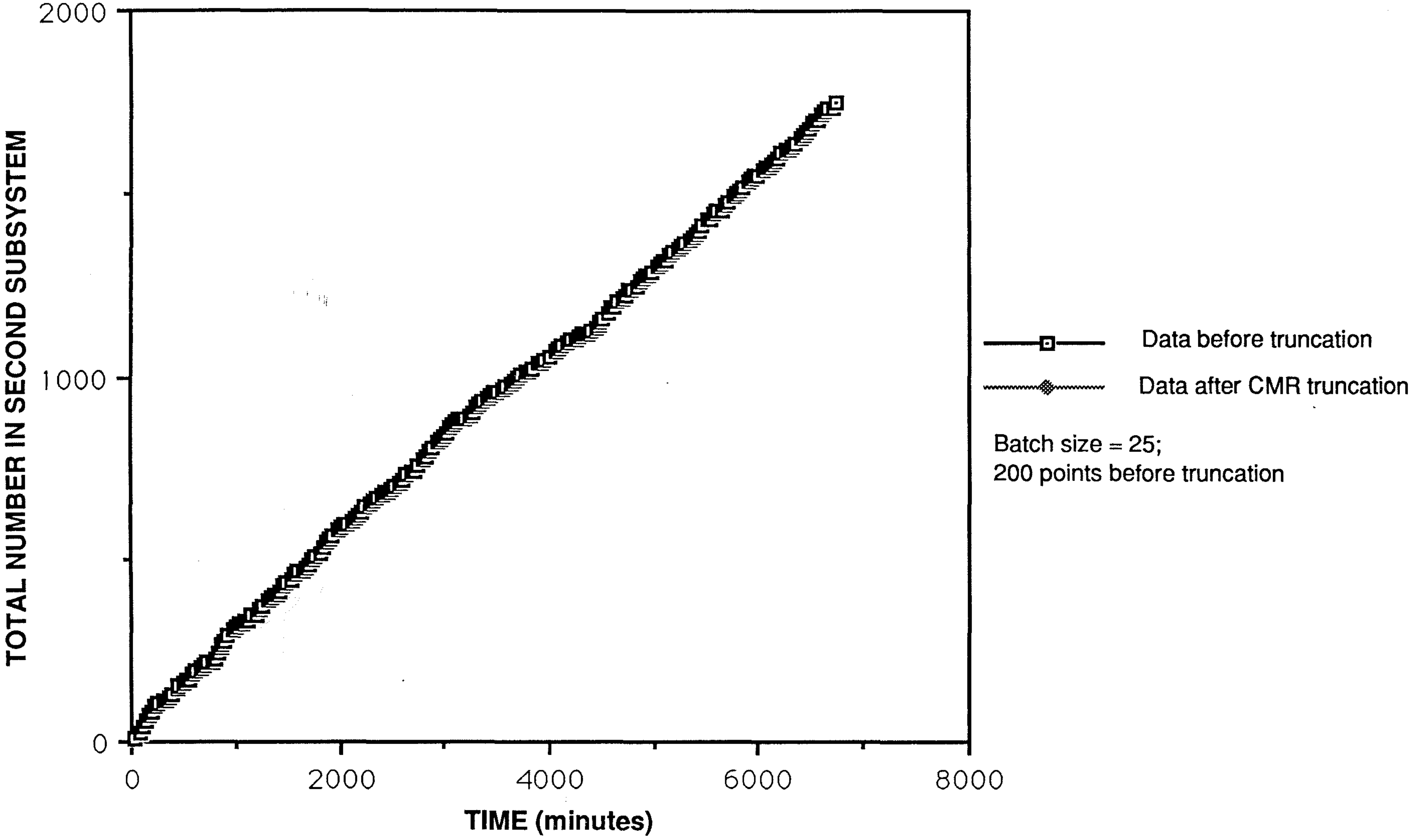
Transient Queue Model : Run 7





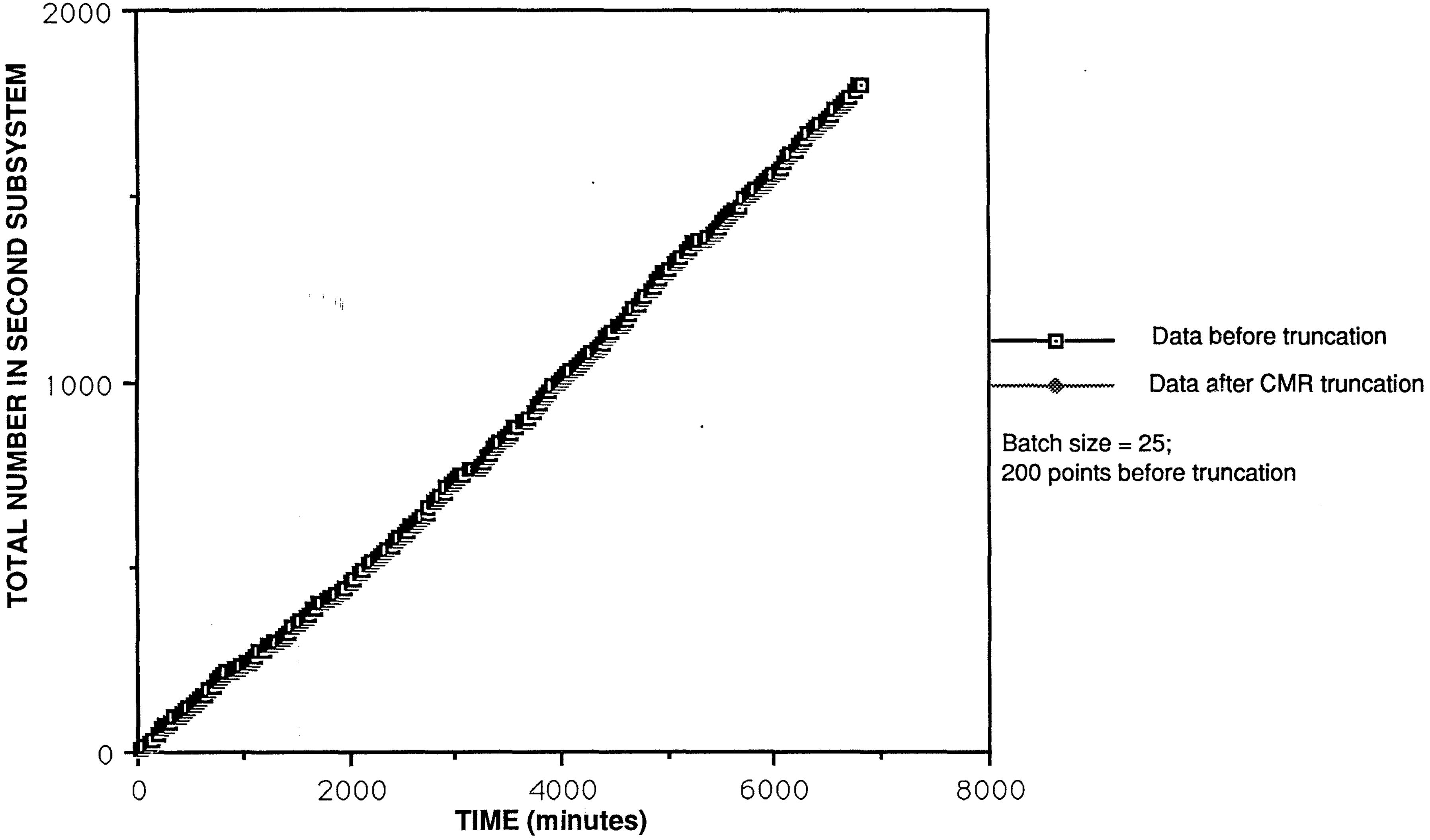
Batch size = 25;
200 points before truncation

Transient Queue Model : Run 9



Batch size = 25;
200 points before truncation

Transient Queue Model : Run 10



Batch size = 25;
200 points before truncation