Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

OUTPUT ANALYSIS FOR SIMULATIONS

Marvin K. Nakayama

Computer Science Department
New Jersey Institute of Technology
Newark, NJ 07102, U.S.A.

ABSTRACT

We discuss methods for statistically analyzing the output from stochastic discrete-event or Monte Carlo simulations. Terminating and steady-state simulations are considered.

1 INTRODUCTION

So you've finally finished developing your simulation model. You spent countless hours developing an understanding of the underlying processes, collecting data, fitting the data to various probability distributions, and coding and debugging your simulation program. You carefully selected a performance measure you felt was appropriate to evaluate the system, and your program outputs an estimate of this measure. You then ran the simulation program once, and the results seemed to indicate that if the system design in your program was actually put into practice, it would perform well. You showed your boss the results, who then gave you the green light to implement this system design. However, once the system was in place, it performed poorly, not at all like the results that you obtained from your one simulation run. What went wrong?

Many simulations include randomness, which can arise in a variety of ways. For example, in a simulation of a manufacturing system, the processing times required at a station may follow a given probability distribution or the arrival times of new jobs may be stochastic. In a bank simulation, customers arrive at random times and the amount of time spent at a teller is stochastic. Future returns in financial simulations are often modeled as random variables.

Because of the randomness in the components driving a simulation, its output is also random, so statistical techniques must be used to analyze the results. The data-analysis methods taught in introductory statistics courses typically assume that the data are independent and identically distributed (i.i.d.) with a normal distribution, but the output data from simulations are often not i.i.d. normal. For example, consider bank customers' waiting times at an automatic teller machine (ATM). If one customer has an unusually long waiting time, then the next customer probably also will, so the waiting times of the two customers are dependent. Moreover, customers arriving during the lunch hour will usually have longer waiting times than customers coming in at other times, so waiting times are not identically distributed throughout the day. Finally, waiting times are always positive and often skewed to the right, with a possible mode at zero, so waiting times are not normally distributed. For these reasons one often cannot analyze simulation output using the classical statistical techniques developed for i.i.d. normal data.

In this tutorial, we will examine some statistical methods for designing and analyzing simulation experiments. In the next section we begin by distinguishing between two types of performance measures: terminating (or transient) and steady-state (or infinite-horizon or long-run). These two types of measures require different statistical techniques to analyze the results, and Section 3 reviews methods for analyzing output from terminating simulations, while Section 4 covers techniques for steady-state simulations. In Section 5 we discuss the estimation of multiple performance measures, and Section 6 briefly covers other methods useful for analyzing simulation output. Some concluding remarks are given in Section 7.

2 PERFORMANCE MEASURES

One of the first steps in any simulation study is choosing the *performance measure(s)* to compute. In other words, what measures will be used to evaluate how "good" the system is? For example, the performance of a queueing system may be measured by its expected number of customers served in a day, or we may use the long-run average daily cost as a measure of the performance of a supply chain.

There are primarily two types of performance measures for stochastic systems, which we now briefly describe:

1-4244-0501-7/06/$20.00 ©2006 IEEE

1. *Transient performance measures*, also known as *terminating* or *finite-horizon* measures, evaluate the system's evolution over a finite time horizon.
2. *Steady-state performance measures* describe how the system evolves over an infinite time horizon. These are also known as *long-run* or *infinite-horizon* measures.

A simulation in which a transient (resp., steady-state) measure is estimated is called a *transient simulation* (resp., *steady-state simulation*). We now describe these concepts in more depth.

2.1 Transient Performance Measures

Definition: A *terminating simulation* is one for which there is a "natural" event B that specifies the length of time in which one is interested for the system. The event B often occurs either at a time point beyond which no useful information is obtained, or when the system is "cleaned out." For example, if we are interested in the performance of a system during the first 10 time units of operation of a day, then B would denote the event that 10 time units of system time have elapsed. If we want to determine the first time at which a queue has at least 8 customers, then B is the event of the first time the queue length reaching 8. (See Law and Kelton 2000, Section 9.3, for more details.)

Since we are interested in the behavior of the system over only a finite time horizon, the "initial conditions" $\mathcal{C}$ (i.e., conditions under which the system starts) can have a large impact on the performance measure. For example, queueing simulations often start with no customers present, which would be the conditions $\mathcal{C}$ in this setting.

In a transient simulation, we have the following

Goal: To compute

$$\mu = E(X), \tag{1}$$

where X is a random variable representing the (random) performance of the system over some finite horizon and E denotes expectation (or average).

We now examine some examples of transient performance measures.

Example 1 Consider a bank vestibule containing an ATM. The vestibule is only open during normal banking business hours, which is 9:00am to 5:00pm, so customers can access the ATM only during those times. Any customers in the vestibule at 5:00pm will be allowed to complete their transactions, but no new customers will be allowed in. Let Z be the number of customers using the ATM in a day, and we may be interested in determining the following terminating performance measures:

- $E[Z]$, the expected value of Z. To put things in the framework of (1), we set $X = Z$.
- $P\{Z \geq 500\} = E[I(Z \geq 500)]$, which is the probability that at least 500 customers use the ATM in a day, where $I(A)$ is the indicator function of an event A, which takes on the value 1 if A occurs, and 0 otherwise. In the notation of (1), $X = I(Z \geq 500)$ in this case.

The initial conditions $\mathcal{C}$ might be that the system starts out empty each day, and the terminating event B is that it is past 5:00pm and there are no more customers in the vestibule.

Alternatively we might define Z to be the average waiting time (in seconds) of the first 50 customers in a day. We can then define the following performance measures:

- $E[Z]$, the expected value of Z. In this case, $X = Z$ in the notation of (1).
- $P\{Z \leq 30\} = E[I(Z \leq 30)]$, which is the probability that the average waiting time of the first 50 customers is no more than 30 seconds. Here, $X = I(Z \leq 30)$ in (1).

In this case we might specify the initial conditions $\mathcal{C}$ to be that the system starts out empty each day, and the terminating event B is that 50 customers have finished their waits in line.

2.2 Steady-State Performance Measures

Now we consider steady-state performance measures. Let $\mathbf{Y} = (Y_1, Y_2, Y_3, \ldots)$ be a (discrete-time) stochastic process representing the output of a simulation. For example, if the vestibule containing the ATM in our previous example is now open 24 hours a day, then Y_i might represent the waiting time of the ith customer since the ATM was installed. Let $F_i(y|\mathcal{C}) = P(Y_i \leq y|\mathcal{C})$ for $i = 1, 2, \ldots$, where as before, $\mathcal{C}$ represents the initial conditions of the system at time 0. Observe that $F_i(\cdot|\mathcal{C})$ is the distribution function of Y_i given the initial conditions $\mathcal{C}$. We are now interested in the behavior of the system over an infinite time horizon, and it is often the case that the effects of the initial conditions $\mathcal{C}$ become negligible after a sufficiently long time has elapsed.

Definition: If

$$F_i(y|\mathcal{C}) \to F(y) \text{ as } i \to \infty \tag{2}$$

for all y and for any initial conditions $\mathcal{C}$, then $F(y)$ is called the *steady-state distribution* of the process $\mathbf{Y}$. If Y is a random variable with distribution F, we say that Y has the steady-state distribution, and we sometimes write this as $Y_i \xrightarrow{\mathcal{D}} Y$ as $i \to \infty$, which is read as "Y_i converges in distribution to Y."

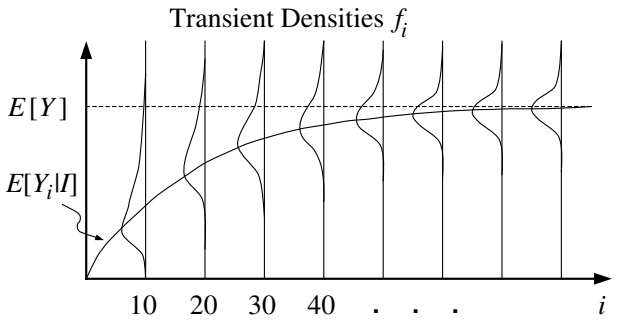

Figure 1: Densities f_i of an Output Process $(Y_1, Y_2, \ldots)$

The interpretation of (2) is that for all i sufficiently large,

$$F_i(y|\mathcal{C}) \approx F(y), \text{ for all } y. \tag{3}$$

The value of i for which the approximation holds depends very much on the particular system being simulated. Note that (3) does not mean that the *values* of the Y_i are all the same for large i, but rather that the *distribution* of Y_i (given the initial conditions $\mathcal{C}$) is close to F for large i. Indeed, the steady-state random variable Y (and also the Y_i for large i) may still have plenty of variability. When Y is a random variable with distribution F, $E(Y)$ is a *steady-state performance measure*. It can be shown under great generality that $E(Y_i|\mathcal{C}) \to E(Y)$ as $i \to \infty$ for all initial conditions $\mathcal{C}$ when (2) holds. Figure 1 gives an example of density functions f_i approaching some limiting density f as i gets larger.

Many systems do not have a steady state. For example, consider our previous example of an ATM that is accessible only during business hours. Let Y_i be the waiting time of the ith customer to arrive since the ATM was installed. Then, the process $\mathbf{Y}$ does not have a steady state because the first customer of each day always has no wait, whereas other customers may have to wait. For example, suppose 500 customers are served on the first day, so day 2 begins with customer 501, who has no wait since there is no one ahead of him on that day. Since this happens every day, (2) cannot hold. On the other hand, if the ATM were accessible 24 a day, then a steady state may exist.

In the above example where the ATM is only available from 9:00am to 5:00pm, we may be able to obtain a process $\mathbf{Y}$ that does have a steady state if we define the Y_i differently. In particular, suppose Y_i is the average waiting time of all the customers on the ith day since the ATM first became operational. Then, $\mathbf{Y}$ may have a steady state. (It still may not if the distribution of the number of customers in a day depends on the particular day of the week, or if there are seasonal variations, in which case (2) cannot hold.)

Example 2 Consider the ATM from before, but now suppose that it accessible all the time. Let Y_i be the number of customers served on the ith day of operation, and suppose that over time, the system "settles down" into steady state; i.e., $Y_i \xrightarrow{\mathcal{D}} Y$ as $i \to \infty$. We now may be interested in determining the following steady-state performance measures:

- $E[Y]$, which is the expected steady-state number of customers served in a day;
- $P\{Y \geq 400\} = E[I(Y \geq 400)]$, which is the steady-state probability that at least 400 customers are served in a day.

Again, we may let the initial conditions $\mathcal{C}$ denote that the system begins operations on the first day with no customers present, and over time, the effects of the initial conditions "wash away."

3 OUTPUT ANALYSIS FOR TRANSIENT SIMULATIONS

We now discuss how to analyze the output from a transient simulation. Recall our goal is to calculate $\mu = E(X)$, where X is a random variable representing the performance of the system over some finite horizon with initial conditions $\mathcal{C}$. The basic approach to estimate μ using simulation is as follows:

Method: Generate $n \geq 2$ i.i.d. replicates of X, say $X_1, X_2, \ldots, X_n$, and form the (point) estimator

$$\bar{X}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{4}$$

We generate i.i.d. replicates of X by running independent simulations of the system under study. We make the replicates independent by using non-overlapping streams of random numbers from the random-number generator. We ensure the replicates are identically distributed by starting each simulation using the same initial conditions $\mathcal{C}$ and using the same dynamics to govern the evolution of the system.

The law of large numbers guarantees $\bar{X}(n) \approx \mu$ for large sample sizes n. But how close is $\bar{X}(n)$ to μ? The central limit theorem (CLT) provides an answer. Specifically, let σ^2 denote the *variance* of X, i.e., $\sigma^2 = \text{Var}(X)$. We sometimes also refer to the *standard deviation* of X, which is $\sigma = \sqrt{\sigma^2}$. Also, let

$$S^2(n) = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \bar{X}(n))^2, \tag{5}$$

which is the *sample variance* of $X_1, \ldots, X_n$, and is an estimator of σ^2. The *sample standard deviation* is $S(n) = \sqrt{S^2(n)}$, which is an estimator of σ. A variant of the standard CLT

asserts that for n large,

$$\frac{\sqrt{n}}{S(n)}\left(\bar{X}(n)-\mu\right) \stackrel{\mathcal{D}}{\approx} N(0,1), \tag{6}$$

where $N(a,b)$ denotes a normal random variable having mean a and variance b and $\stackrel{\mathcal{D}}{\approx}$ means "has approximately the same distribution as." The approximation in (6) is usually reasonable for $n \geq 50$, and it becomes exact as $n \to \infty$.

We now use (6) to derive a *confidence interval* for μ. First define the *confidence level* $1-\delta$ with $0 < \delta < 1$; typically, one chooses $\delta = 0.1$, 0.05 or 0.01. Then, we look up in a z-table the constant $z \equiv z_{1-\delta/2}$ for which $P\{N(0,1) \leq z\} = 1-\delta/2$; e.g., $z = 1.65$ when $\delta = 0.1$, $z = 1.96$ when $\delta = 0.05$, and $z = 2.58$ when $\delta = 0.01$. Virtually any introductory statistics book provides a z-table; also see Table T.1 of Law and Kelton (2000) or Table A.3 of Banks et al. (2001). Then

$$\begin{aligned} 1-\delta &= P\{-z \leq N(0,1) \leq z\} \\ &\approx P\left\{-z \leq \frac{\sqrt{n}}{S(n)}(\bar{X}(n)-\mu) \leq z\right\} \qquad (7) \\ &= P\left\{\mu \in \left[\bar{X}(n) \pm \frac{zS(n)}{\sqrt{n}}\right]\right\}, \qquad (8) \end{aligned}$$

where the approximation in (7) follows for large n from (6). Note that (8) implies that when n is large, the interval

$$\left[\bar{X}(n) - \frac{zS(n)}{\sqrt{n}},\ \bar{X}(n) + \frac{zS(n)}{\sqrt{n}}\right] \tag{9}$$

has roughly probability $1-\delta$ of containing the true mean μ, and we call the interval in (9) an approximate $100(1-\delta)\%$ confidence interval for μ. Thus, we arrive at the following:

Procedure to construct confidence intervals for transient measure μ:

1. Specify a confidence level $1-\delta$ with $0 < \delta < 1$ and a sample size n that is large. Also, look up in a z-table the value of z such that $P\{N(0,1) \leq z\} = 1-\delta/2$. Typically, one chooses $\delta = 0.1$, 0.05 or 0.01, and one should choose $n \geq 50$.
2. Generate n i.i.d. replicates $X_1, X_2, \ldots, X_n$ of X.
3. Using the n data points $X_1, X_2, \ldots, X_n$, calculate the sample mean $\bar{X}(n)$ using (4) and the sample variance $S^2(n)$ using (5).
4. Use (9) to construct an approximate $100(1-\delta)\%$ confidence interval for μ.

An interpretation of the approximate $100(1-\delta)\%$ confidence interval for μ in (9) is that we are highly confident (i.e., approximately $100(1-\delta)\%$ confident) that the true mean μ lies in the interval (9). Thus, a confidence interval provides a form of error bounds for our estimator $\bar{X}(n)$ of μ. The half width H_n of the confidence interval in (9) is

$$H_n = \frac{zS(n)}{\sqrt{n}}, \tag{10}$$

i.e., the confidence interval in (9) is $\bar{X}(n) \pm H_n$. It can be shown that $S(n) \approx \sigma$ for large n, so as the sample size n increases, the half width decreases at rate $1/\sqrt{n}$. In particular, this means that to obtain one additional significant figure of accuracy (i.e., increase accuracy by a factor of 10), we need to increase the sample size n by a factor of 100. Thus, the estimator $\bar{X}(n)$ converges to μ rather slowly.

If we construct the confidence interval (9) using the above steps, the probability is approximately $1-\delta$ that the interval will contain μ. In other words, if we repeat these steps m independent times, this will give us m different confidence intervals. Some of them will contain (cover) μ, and others will not. The theory says that approximately $(1-\delta)m$ of the m intervals should cover μ. For example, if we constructed $m = 1000$ independent 95% confidence intervals, we would expect that about 950 of them would contain μ, while about 50 would not. In practice, though, this does not always happen. The approximation in our CLT (6) only becomes exact as the sample size $n \to \infty$, so the coverage is only approximately $1-\delta$ for large but finite n, i.e.,

$$\begin{aligned} &P\left\{\mu \in \left[\ \bar{X}(n) - \frac{zS(n)}{\sqrt{n}},\ \bar{X}(n) + \frac{zS(n)}{\sqrt{n}}\ \right]\right\} \\ &\quad \approx \quad 1-\delta. \end{aligned} \tag{11}$$

The true probability that μ lies in the interval in (9) is known as the *coverage*.

It would be nice to know when the approximation in (11) is good, and when it is not. It turns out that the quality of the CLT approximation in (6) is largely influenced by the value of the *skewness* of X. The more symmetric the density of X is, the better the CLT approximation in (6) is, which leads to (11) being more accurate. If the density of X is highly asymmetric (as is typical of queueing simulations), the CLT approximation is not so good, and the coverage of the confidence interval in (9) may be significantly less than $1-\delta$. In fact, it is not unusual for confidence intervals that are supposed to have 90% coverage to actually only have, say, 75% coverage. See p. 257 of Law and Kelton (2000) for more discussion.

3.1 Pre-specifying Confidence Interval Widths

In the previous section we discussed so-called *fixed-sample-size methods* for estimating a transient performance measure $\mu = E[X]$, where X represents the random performance of

the system over some finite time horizon. These methods are so named because the sample size is fixed prior to running any simulations. However, before executing a simulation, we usually do not know how large the resulting half width (10) will be since it depends on the output generated. In many situations, though, we would like to end up with an estimator with a small prespecified error ε, i.e., we want the $100(1-\delta)\%$ confidence interval to be $\bar{X}(n) \pm \varepsilon$.

Example 3 Suppose we want to estimate the expected daily withdrawals from an ATM. If we want the estimator to be within \$500 of the correct value with confidence level $1-\delta$, then we set the desired (absolute) error to be $\varepsilon = 500$.

To achieve our goal of having a confidence interval with half width ε, we set H_n in (10) equal to ε and solve for n, yielding

$$n = \left(\frac{zS(n)}{\varepsilon} \right)^2. \tag{12}$$

This suggests that if we take n samples, where n is determined by (12), then the resulting confidence interval should have half width that is approximately ε.

We now describe a two-stage procedure to construct a confidence interval with half width that is roughly a prespecified value ε. In the first stage we generate n_0 trial runs and compute the sample standard deviation $S(n_0)$, and then we substitute $S(n_0)$ into the right-hand side of (12) to compute the total sample size required. The following is a variation of a procedure developed by Stein (1945).

Two-stage procedure for absolute-precision confidence intervals:

1. Select n_0, a sample size for the set of trial runs. (In practice, one should specify $n_0 \geq 50$). Also, select the desired error ε. (In practice, one should specify ε to be "small," the meaning of which depends on the context.) Also, specify a confidence level $1-\delta$ with $0 < \delta < 1$, and look up in a z-table the value of z such that $P\{N(0,1) \leq z\} = 1 - \delta/2$.
2. Generate n_0 (independent) *pilot* runs, yielding samples $X_1, X_2, \ldots, X_{n_0}$.
3. Calculate the sample variance

$$S_1^2(n_0) = \frac{1}{n_0 - 1} \sum_{i=1}^{n_0} (X_i - \bar{X}(n_0))^2$$

of the pilot runs $X_1, X_2, \ldots, X_{n_0}$, where $\bar{X}(n_0) = (1/n_0) \sum_{i=1}^{n_0} X_i$.

4. Calculate

$$N_a(\varepsilon) = \left\lceil \left(\frac{zS_1(n_0)}{\varepsilon} \right)^2 \right\rceil,$$

where $\lceil x \rceil$ is the "round up" function.

5. Generate $N_a(\varepsilon)$ (independent) *production* runs that are independent of $X_1, X_2, \ldots, X_{n_0}$. The samples from the production runs are denoted $X_{n_0+1}, X_{n_0+2}, \ldots, X_{n_0+N_a(\varepsilon)}$.
6. Set

$$\widetilde{X}(\varepsilon) = \frac{1}{N_a(\varepsilon)} \sum_{j=n_0+1}^{n_0+N_a(\varepsilon)} X_j$$

and

$$\widetilde{S}^2(\varepsilon) = \frac{1}{N_a(\varepsilon) - 1} \sum_{j=n_0+1}^{n_0+N_a(\varepsilon)} (X_j - \widetilde{X}(\varepsilon))^2,$$

which are the sample mean and sample variance of only the values from the production runs.

7. Then

$$\left[\widetilde{X}(\varepsilon) - \frac{z\widetilde{S}(\varepsilon)}{\sqrt{N_a(\varepsilon)}},\ \widetilde{X}(\varepsilon) + \frac{z\widetilde{S}(\varepsilon)}{\sqrt{N_a(\varepsilon)}} \right]$$

is an approximate $100(1-\delta)\%$ confidence interval for μ, the half-width of which should be *approximately* ε.

If ε is small (as is usual in applications), then $N_a(\varepsilon) \gg n_0$ (i.e., $N_a(\varepsilon)$ will be much larger than n_0) so that throwing away the first n_0 pilot observations in forming the estimators $\widetilde{X}(\varepsilon)$ and $\widetilde{S}^2(\varepsilon)$ is not going to affect the procedure much.

The previous procedure results in an absolute-precision confidence interval, but in many contexts, one desires relative-precision intervals. For example, we may want our confidence interval to be $\pm 5\%$ of the point estimator. To achieve this, we change the total sample size from $N_a(\varepsilon)$ to

$$N_r(\varepsilon) = \left\lceil \left(\frac{z\widetilde{S}_1(n_0)}{\bar{X}(n_0)\varepsilon} \right)^2 \right\rceil,$$

where $\bar{X}(n_0)$ is the first-stage sample mean and ε is the desired relative precision. For example, for a confidence interval that is $\pm 5\%$, set $\varepsilon = 0.05$.

4 OUTPUT ANALYSIS FOR STEADY-STATE SIMULATIONS

We now discuss the estimation of steady-state performance measures. There are two cases to consider:

1. Discrete-time process: $\mathbf{Y} = (Y_i : i = 1, 2, \ldots)$ is an output process with an integer-valued time index, and our goal is to estimate (and produce confidence

intervals for) ν, where ν is defined such that

$$\frac{1}{m}\sum_{i=1}^{m} Y_i \to \nu \tag{13}$$

as $m \to \infty$.

2. Continuous-time process: $\mathbf{Y} = (Y(s) : s \geq 0)$ is an output process with a continuous-valued time index, and we want to estimate (and produce confidence intervals for) ν, where ν is defined such that

$$\frac{1}{s}\int_0^s Y(u)\,du \to \nu \tag{14}$$

as $s \to \infty$.

We previously saw in Section 2.2 some examples of steady-state measures for a discrete-time process. For example, Y_i could be the waiting time of the ith customer to a queueing system, so ν represents the steady-state expected waiting time. We now give an example of a continuous-time process.

Example 4 Suppose that the ATM from before is accessible 24 hours a day, and let $Y(s)$ denote the number of customers waiting in line at time s. We define the continuous-time stochastic process $\mathbf{Y} = (Y(s) : s \geq 0)$, and assuming that $\mathbf{Y}$ has a steady state (which would not be the case if the distribution of the number of customers waiting depends on the time of day), then we may be interested in calculating ν defined in (14), which in this case is the long-run time-average number of customers waiting. Another possible measure is

$$\lim_{s\to\infty} \frac{1}{s}\int_0^s I(Y(u) \geq a)\,du,$$

which is the long-run fraction of time that at least a customers are waiting.

4.1 The Difficulties of Output Analysis of Steady-State Simulations

We will concentrate on discrete-time processes (continuous-time processes can be handled in a similar manner). Our goal is to estimate and produce confidence intervals for the steady-state parameter ν. First, we examine how to produce a point estimator for ν. As we can see in (13), the parameter ν can be viewed as the long-run average level of Y_i. Thus, if we set

$$\bar{Y}(m) = \frac{1}{m}\sum_{i=1}^{m} Y_i,$$

then

$$\bar{Y}(m) \approx \nu$$

for large sample sizes m. In other words, running a "long" simulation (i.e., taking m large) will result in an estimator $\bar{Y}(m)$ that is "close" to ν. Hence, the problem of constructing an estimator for ν is easily solved.

However, the task of constructing a confidence interval for ν is more delicate. For virtually all reasonably behaved systems possessing a unique steady state, one can show that a central limit theorem for $\bar{Y}(m)$ is valid; i.e., there exists a constant $\bar{\sigma}$ such that

$$\frac{\sqrt{m}}{\bar{\sigma}}(\bar{Y}(m) - \nu) \stackrel{\mathcal{D}}{\approx} N(0,1) \tag{15}$$

for m sufficiently large.

Definition: The parameter $\bar{\sigma}^2$ is called the *time-average variance constant* of the steady-state simulation.

Unfortunately, it is not so straightforward to use the CLT in (15) to construct a confidence interval for ν. The problem lies in the fact that it is a non-trivial matter to estimate $\bar{\sigma}$ (or equivalently $\bar{\sigma}^2$). The sample variance $S^2(n)$ in (5) used to estimate σ^2 in the transient-simulation setting is only valid for i.i.d. data. In steady-state simulations, $Y_1, Y_2, \ldots$ are typically not i.i.d. Thus, we cannot use (5) applied to the $Y_1, Y_2, \ldots$ to estimate $\bar{\sigma}^2$.

4.2 Method of Multiple Replications

The *method of multiple replications* offers one escape from this difficulty of estimating $\bar{\sigma}$. Suppose that rather than simulating one long replicate of length m, we simulate r *independent* and identically distributed replications, each of length $k = m/r$. We should choose r small, say $10 \leq r \leq 30$, so that the length k of each replication is large. We need k large since we are interested in the long-run behavior of the process $\mathbf{Y}$. We achieve independence of the replications by using non-overlapping streams of random numbers for the different replications. We obtain identically distributed replications by starting each with the same initial conditions and using the same system dynamics to generate each replication. Because we now have r independent observations, we can form a sample variance across the replications. This is the basic idea underlying the method of multiple replications.

Suppose that we have run r i.i.d. replications, each having run length k, and the output from all the simulations

is

$$\begin{array}{ccccc} Y_{1,1} & Y_{1,2} & Y_{1,3} & \cdots & Y_{1,k}, \\ Y_{2,1} & Y_{2,2} & Y_{2,3} & \cdots & Y_{2,k}, \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{r,1} & Y_{r,2} & Y_{r,3} & \cdots & Y_{r,k}. \end{array} \tag{16}$$

The entries in the first row are the k observations from the first replication, the entries in the second row are the k observations from the second replication, and so on. Now let X'_j be the average of the entries in the jth row; i.e.,

$$X'_j = \frac{1}{k}\sum_{i=1}^{k} Y_{j,i}.$$

Thus, X'_1 is the average of the observations in the first row of (16), X'_2 is the average of the observations in the second row of (16), and so on. Then $X'_1, X'_2, \ldots, X'_r$ are i.i.d. observations with $E(X'_j) \approx \nu$ for each $j = 1, 2, \ldots, r$, if k is sufficiently large by virtue of (13). So we can use classical statistics to form a point estimator and confidence interval using the observations $X'_1, X'_2, \ldots, X'_r$. Specifically, let

$$\bar{X}'(r) = \frac{1}{r}\sum_{j=1}^{r} X'_j$$

and

$$S'^2(r) = \frac{1}{r-1}\sum_{j=1}^{r} (X'_j - \bar{X}'(r))^2$$

be the sample mean and sample variance, respectively, of the X'_j. Then, an approximate $100(1-\delta)\%$ confidence interval for ν is given by

$$\left[\bar{X}'(r) - \frac{tS'(r)}{\sqrt{r}},\ \bar{X}'(r) + \frac{tS'(r)}{\sqrt{r}}\right],$$

where $t \equiv t_{r-1,1-\delta/2}$ is chosen such that $P\{T_{r-1} \le t\} = 1-\delta/2$ and T_{r-1} is a Student-t random variable with $r-1$ degrees of freedom. Virtually all introductory statistics books provide t-tables giving values of t for various δ and degrees of freedom; also see Table T.1 of Law and Kelton (2000) or Table A.5 of Banks et al. (2001). (Here we use the critical point from a t-distribution rather than a standard normal distribution because the number r of replications is often small.)

A major problem with the method of multiple replications is that, while the technique permits simple estimation of the variance, the multiple-replicate estimator $\bar{X}'(r)$ can be significantly contaminated by the presence of *initialization bias*. Specifically, the law of large numbers guarantees that

$$X'_j = \frac{1}{k}\sum_{i=1}^{k} Y_{j,i} \approx \nu$$

for large k. However, since each replicate is typically started with initial conditions $\mathcal{C}$ that are atypical of the steady state (e.g., queueing simulations are often started with no customers present), it often follows that for any finite k,

$$E\left[\frac{1}{k}\sum_{i=1}^{k} Y_{j,i}\right] \neq \nu.$$

Thus, we conclude that if the number of replicates r is large relative to the run length k of each replication, then the estimator $\bar{X}'(r)$ may be significantly biased by the initial conditions.

A partial solution to this problem is to use *initial-data deletion*, which we now describe. Suppose that we somehow can determine the first c observations of the simulation are significantly contaminated, i.e., not very representative of steady state. Also, suppose all observations Y_i with $i > c$ are not significantly contaminated. Then in each replication, we will delete the first c observations when calculating the sample mean of the replication. Specifically, for each replication $j = 1, 2, \ldots, r$, let

$$X_j = \frac{1}{k-c}\sum_{i=c+1}^{k} Y_{j,i}$$

be the sample mean of the (non-contaminated) observations $Y_{j,c+1}, Y_{j,c+2}, \ldots, Y_{j,k}$, in replication j. After simulating the r replications, compute

$$\bar{X}(r) = \frac{1}{r}\sum_{j=1}^{r} X_j$$

and

$$S^2(r) = \frac{1}{r-1}\sum_{j=1}^{r} (X_j - \bar{X}(r))^2,$$

which are the sample mean and sample variance, respectively, of the X_j. Then, an approximate $100(1-\delta)\%$ confidence interval for ν is given by

$$\left[\bar{X}(r) - \frac{tS(r)}{\sqrt{r}},\ \bar{X}(r) + \frac{tS(r)}{\sqrt{r}}\right].$$

For more details on initial-data deletion, including some heuristics to determine c, see Section 9.1 of Law and Kelton (2000).

One problem with initial-data deletion is that in each of the r replications, we have to delete c observations. Thus, we are throwing away a total of rc observations over all of the replications. If we used a *single-replicate algorithm* (i.e., one with $r = 1$), then we would only delete a total of c observations.

4.3 Single-Replicate Methods

Typically in practice, two observations Y_i and Y_{i+p} are almost independent when p is large, for each i. For example, suppose Y_i is the waiting time of the ith customer in a queueing system. Then we would expect that the waiting time of the 100th customer to be almost independent of the 10th customer's waiting time.

Now suppose that we run a single simulation of length m, giving us observations $Y_1, Y_2, \ldots, Y_m$. Suppose we group the m observations into n large, non-overlapping *batches*, each of size b (so $m = nb$), where the first batch consists of the first b observations, the second batch consists of the next b observations, and so on:

$$\underbrace{Y_1\ Y_2\ \cdots\ Y_b}_{\text{Batch 1}}\ \underbrace{Y_{b+1}\ Y_{b+2}\ \cdots\ Y_{2b}}_{\text{Batch 2}}\ \underbrace{Y_{2b+1}\ Y_{2b+2}\ \cdots\ Y_{3b}}_{\text{Batch 3}}\ \cdots$$

If b is chosen to be large, then most of the observations in one batch should be almost independent of most of the observations in any other batch. Essentially the only dependence that exists is between observations in two adjacent batches. Observations in batches that are not adjacent are almost independent. Moreover, if we compute the sample mean of each of the batches, then the sample means should be almost independent when the batch size b is large. Also, each sample mean will be close to normally distributed for large b, since it is a sample mean and so it satisfies a CLT (see (15)). Using the above observations, we now present the following:

Method of batch means to construct confidence intervals in steady-state simulations:

1. Select a total run length m, which is large. Also, select a number of batches n. (Schmeiser 1982 suggests choosing $10 \le n \le 30$.)
2. Run a simulation generating a total of m observations. This results in observations $Y_1, Y_2, \ldots, Y_m$.
3. Then group the m observations into n batches, each of size $b = m/n$. For $j = 1, 2, \ldots, n$, the jth *batch mean* is calculated as

$$\bar{Y}_j(b) = \frac{1}{b} \sum_{l=(j-1)b+1}^{jb} Y_l,$$

which is the sample average of the observations in the jth batch. Note that $\bar{Y}_j(b)$ is the sample mean of the b observations in the jth batch.

4. We then treat $\bar{Y}_1(b), \bar{Y}_2(b), \ldots, \bar{Y}_n(b)$ as i.i.d. observations (note that they are not, but should be reasonably close for large batch sizes b) and use classical statistics to construct a confidence interval. Specifically, compute

$$\bar{\bar{Y}}(n,b) = \frac{1}{n} \sum_{j=1}^{n} \bar{Y}_j(b) = \frac{1}{m} \sum_{i=1}^{m} Y_i,$$

and

$$S^2(n,b) = \frac{1}{n-1} \sum_{j=1}^{n} \left(\bar{Y}_j(b) - \bar{\bar{Y}}(n,b) \right)^2$$

as the sample mean and sample variance, respectively, of the n batch means, and an approximate $100(1-\delta)\%$ confidence interval for ν is

$$\left[\bar{\bar{Y}}(n,b) - \frac{tS(n,b)}{\sqrt{n}},\ \bar{\bar{Y}}(n,k) + \frac{tS(n,b)}{\sqrt{n}} \right],$$

where $t \equiv t_{n-1,1-\delta/2}$ is chosen such that $P\{T_{n-1} \le t\} = 1 - \delta/2$ for T_{n-1} a Student-t random variable with $n-1$ degrees of freedom.

We can easily modify the above procedure to incorporate initial-data deletion by instead collecting a total of $m + c$ observations and removing the first c contaminated observations. Then apply the method of batch means with the remaining m data points. When using a single-replicate method such as batch means, we only need to delete a total of c observations to apply initial-data deletion, as opposed to rc when using the method of multiple replications with r replications. Whitt (1991) provides a mathematical analysis that basically yields the following:

Rule of thumb: Single replicate procedures tend to be better (as measured by the mean square error of the steady-state estimator) than multiple-replicate procedures.

There has been a lot of recent work on improvements to the batch-means method described above. See Schmeiser and Song (1996) for a survey.

4.4 Other Methods

There are numerous other methods for statistically analyzing simulation output in the steady-state context. These include spectral (e.g., Anderson 1994), regenerative (Crane and Iglehart 1975, Shedler 1993), and standardized time series methods (Schruben 1983), but these techniques require more

sophisticated mathematics to understand and can be more difficult to implement. For an overview of these other techniques, see Bratley, Fox and Schrage (1987) or Law and Kelton (2000). Finally, Nakayama (1994) presents two-stage procedures for obtaining fixed-width confidence intervals in steady-state simulations.

5 ESTIMATING MULTIPLE PERFORMANCE MEASURES

Consider our previous example of an ATM that is accessible only between 9:00am and 5:00pm, and suppose that we want to compute

- μ_1, the expected number of customers served in a day;
- μ_2, the probability that the number served in a day is at least 1000;
- μ_3, the expected amount of money withdrawn from the ATM in a day.

These are all transient performance measures, and suppose we use the same simulation to estimate all 3 measures by running n independent replications. Let $X_{1,i}$ denote the number of customers served in the ith replication. Let $X_{2,i}$ be 1 if at least 1000 customers are served in the ith replication, and 0 otherwise. Let $X_{3,i}$ be the amount of money withdrawn on the ith replication.

After running n replications, suppose we construct a 95% confidence interval for each μ_s, $s = 1,2,3$. Let I_s denote the 95% confidence interval for μ_s, so if we ran a sufficiently large number n of replications, then $P\{\mu_s \in I_s\} \approx 0.95$ for each $s = 1,2,3$. But what can we say about the *joint* coverage of the 3 confidence intervals; i.e., what is $P\{\mu_s \in I_s, \text{ for all } s = 1,2,3\}$?

More generally, suppose that we are estimating q means μ_s, $s = 1,2,\ldots,q$, and for each μ_s, we construct a $100(1-\delta_s)\%$ confidence interval I_s. What can we say about $P\{\mu_s \in I_s, \text{ for all } s = 1,2,\ldots,q\}$? In general, it is difficult to determine the joint confidence level, but Bonferroni's inequality provides a lower bound for this probability:

$$P\{\mu_s \in I_s, \text{ for all } s = 1,2,\ldots,q\} \geq 1 - \sum_{s=1}^{q} \delta_s.$$

Thus, in our previous example in which we had three 95% confidence intervals, the Bonferroni inequality implies the joint probability that all three confidence intervals contain their respective true means is at least 85%. Therefore, our joint confidence level for all three intervals is less than the confidence level for any single interval. If we want the joint confidence to be at least 95%, then we might set $\delta_s = 0.01$ for each s. This would yield individual 99% confidence intervals, with the joint probability being at least 0.97. Thus, to have high confidence that all of our individual confidence intervals contain their respective means, we need to construct the individual confidence intervals with even higher confidence levels.

Often, one wants to compare different systems to see which one is the "best." For example, we may have 5 possible designs for a manufacturing system, and we want to determine which has the highest expected daily production. There is substantial literature on this topic, much of it in the areas of so-called *selection procedures* and *multiple-comparison procedures*. For an overview of these and other simulation-optimization methods, see Fu, Glover and April (2005).

6 OTHER USEFUL METHODS

We now briefly discuss some other techniques that can be useful for simulations. *Variance-reduction techniques* (VRTs), which are also known as *efficiency-improvement techniques*, can lead to simulation estimators with smaller error (variance) by typically either collecting additional information from the simulation run(s) or changing or controlling the way in which the simulation is run. Some of the more widely used VRTs include the following:

- *Common random numbers* (e.g., see Section 11.2 of Law and Kelton 2000) can improve simulations comparing two or more systems by running the simulations of the various systems using the same stream of (uniform) random numbers. In general this leads to fairer comparisons in the sense that all systems are subjected to the same random inputs. This generally induces positive correlation among the resulting estimators, which can be advantageous when estimating differences of performance measures between systems.
- *Antithetic variates* (e.g., see Section 11.3 of Law and Kelton 2000) can improve results from simulating a single system by inducing negative correlations between pairs of replications.
- The method of *control variates* (e.g., see Section 11.4 of Law and Kelton 2000) collects additional data during the simulation, where the mean of the extra collected data is known before running the simulation. For example, in a queueing simulation, one often knows the mean of the service-time distribution, and so one might additionally collect the random service times that are generated during the simulation. The data collected typically is correlated with the simulation output, and this correlation can be exploited to obtain an estimator with lower variance than the standard estimator.
- *Importance sampling* (Hammersley and Handscomb 1965, Glynn and Iglehart 1989) is often

used in rare-event simulations, such as for analyzing buffer overflows in communication networks and system failures of fault-tolerant systems. In these settings, the event of interest, typically some kind of failure, occurs very rarely, and importance sampling changes the dynamics of the system to cause the event to occur more frequently. Unbiased estimators are recovered by multiplying by a correction factor known as the likelihood ratio. Heidelberger (1995) and Nicola, Shahabuddin and Nakayama (2001) review importance-sampling methods for rare-event simulations of queueing and reliability systems.

Other VRTs include stratified sampling, conditional Monte Carlo, and splitting. These and other methods are described in Chapter 11 of Law and Kelton (2000) and Chapter 2 of Bratley, Fox and Schrage (1987).

One is often interested in estimating derivatives of performance measures with respect to system parameters. For example, in a reliability system, one may want to know how the mean time to system failure changes as a component's failure rate varies. This information can be useful in designing systems by identifying components on which to focus to improve overall performance. Also, derivative information can be used with some simulation-optimization methods (e.g., Andradóttir 1998). Techniques for estimating derivatives using simulation include perturbation analysis (Glasserman 1991, Ho and Cao 1991, Fu and Hu 1997) and the likelihood-ratio or score-function method (Reiman and Weiss 1989, Rubinstein 1989, Glynn 1990).

7 CONCLUSIONS

We have described some techniques for statistically analyzing the output from a simulation. It is important to keep in mind that the methods presented here are all *asymptotically* valid, so large run lengths are needed to ensure that valid inferences are drawn.

In addition to the references given throughout the paper, other resources covering simulation-output analysis include Banks (1998), Banks et al. (2001), Fishman (2001), Melamed and Rubinstein (1998), and Ross (2002).

REFERENCES

Anderson, T. W. 1994. *The statistical analysis of time series*. New York: Wiley.

Andradóttir, S. 1998. Simulation optimization. Chapter 9. In *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, Ed. J. Banks. New York: John Wiley and Sons.

Banks, J. 1998. *Handbook of simulation: principles, methodology, advances, applications, and practice*. New York: John Wiley and Sons.

Banks, J., J. S. Carson, II, B. L. Nelson, and D. M. Nicol. 2001. *Discrete-event system simulation*. 3rd edition. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Bratley, P., B. L. Fox, and L. E. Schrage. 1987. *A guide to simulation*. Second Edition. New York: Springer-Verlag.

Crane, M. and D. L. Iglehart. 1975. Simulating stable stochastic systems, III: Regenerative processes and discrete-event simulations. *Operations Research* 23: 33–45.

Fishman, G. S. 2001. *Discrete-event simulation: modeling, programming, and analysis*. New York: Springer-Verlag.

Fu, M., F. Glover and J. April. 2005. Simulation optimization: a review, new developments, and applications. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 83–95.

Fu, M. and J. Q. Hu. 1997. *Conditional Monte Carlo: gradient estimation and optimization applications*. Boston: Kluwer Academic Publishers.

Glasserman, P. 1991. *Gradient estimation via perturbation analysis*. Boston: Kluwer Academic Publishers.

Glynn, P. W. 1990. Likelihood ratio derivative estimators for stochastic systems. *Communications of the ACM* 33: 75–84.

Glynn, P. W. and D. L. Iglehart. 1989. Importance sampling for stochastic simulations. *Management Science* 35: 1367–1392.

Hammersley, J. M. and D. C. Handscomb. 1964. *Monte Carlo methods*. London: Methuen.

Heidelberger, P. 1995. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation* 5: 43–85.

Ho, Y. C. and X. R. Cao. 1991. *Discrete event dynamic systems and perturbation analysis*. Boston: Kluwer Academic Publishers.

Law, A. M. and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3rd edition. New York: McGraw-Hill.

Melamed, B. and R.Y. Rubinstein. 1998. *Modern simulation and modeling*. New York: John Wiley and Sons, Inc.

Nakayama, M. K. 1994. Two-stage stopping procedures based on standardized time series. *Management Science* 40: 1189–1206.

Nicola, V. F., P. Shahabuddin and M. K. Nakayama. 2001. Techniques for fast simulation of models of highly dependable systems. *IEEE Transactions on Reliability* 50: 246–264.

Reiman, M. I. and A. Weiss. 1989. Sensitivity analysis for simulations via likelihood ratios. *Operations Research* 37: 830–844.

Ross, S. M. 2002. *Simulation*. 3rd Edition. Boston: Academic Press.

Rubinstein, R. Y. 1989. Sensitivity analysis and performance extrapolation for computer simulation models. *Operations Research* 37: 72–81.

Schmeiser, B. W. 1982. Batch size effects in the analysis of simulation output. *Operations Research* 30: 556–568.

Schmeiser, B. W., W. T. Song. 1996. Batching methods in simulation output analysis: what we know and what we don't. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. M. Morrice, D. T. Brunner, and J. J. Swain, 122–127.

Schruben, L. W. 1983. Confidence interval estimation using standardized time series. *Operations Research* 31: 1090–1108.

Shedler, G. S. 1993. *Regenerative stochastic simulation*. San Diego: Academic Press.

Stein, C. 1945. A two-sample test for a linear hypothesis whose power is independent of the variance. *Annals of Mathematical Statistics* 16: 243–258.

Whitt, W. 1991. The efficiency of one long run versus independent replications in steady-state simulation. *Management Science* 37: 645–666.

AUTHOR BIOGRAPHY

MARVIN K. NAKAYAMA is an associate professor in the Department of Computer Science at the New Jersey Institute of Technology. He received a Ph.D. in operations research from Stanford University. He won second prize in the 1992 George E. Nicholson Student Paper Competition sponsored by INFORMS and is a recipient of a CAREER Award from the National Science Foundation. He is the stochastic models area editor for *ACM Transactions on Modeling and Computer Simulation* and an associate editor for *INFORMS Journal on Computing*. His research interests include applied probability, statistics, simulation and modeling. His e-mail address is <marvin@njit.edu>, and his web page is <web.njit.edu/~marvin>.