Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

CONTROL OF INITIALIZATION BIAS IN QUEUEING SIMULATIONS USING QUEUEING APPROXIMATIONS

Manuel D. Rossetti

Department of Systems Engineering
University of Virginia
Thornton Hall
Charlottesville, VA 22903

CPT Patrick J. Delaney

Department of Systems Engineering
United States Military Academy
West Point, New York 10996-1779

ABSTRACT

This paper investigates the use of analytical queueing approximations to assist in mitigating the effects of the initial transient period in steady state GI/G/m queueing simulations. We investigate using queueing approximations to stochastically set the initial conditions of the simulation and we develop a new set of truncation heuristics based on GI/G/m queueing approximations. The new truncation heuristics are based on finding the truncation point in the simulation sample path which minimizes the mean squared error of the point estimator. Given that an approximation can be found, our methodology reduces the need for pilot runs and can easily be incorporated into a simulation, with significant results. We present the performance of the heuristics for replication deletion. The adaption to batch means is discussed. The result of our methodology is a less biased and less variable estimator of the expected wait time in the queue. We used the Extend^{TM} simulation package for this research on a Macintosh Quadra 840AV. **Key Words: Start-up Techniques, Output Analysis, Truncation Heuristics, Queueing Simulation**

1 INTRODUCTION

Suppose we are interested in estimating a steady state mean performance parameter, $\theta = E[Y]$, where $Y_1, Y_2, \ldots$ is the output sequence of steady state simulation such that $Pr\{Y_i \leq y\} = F_i(y) \rightarrow F(y) = Pr\{Y \leq y\}$ as $i \rightarrow \infty$. We know that each $F_i(y)$ is really dependent on the initial conditions of the simulation. When we are interested in steady state performance these unrepresentative $F_i(y)$ are termed **transient** distributions. The problem of determining when $F_i(y)$ has become representative of $F(y)$, i. e. when the effects of the initial conditions have worn off, has a long history in the simulation community, see for example Wilson and Pritsker (1978a). The period over which the output process is experiencing the transient distributions is known as the **warm-up period**. Under appropriate ergodic assumptions, we know that the sample mean, $\bar{Y}$, will approach θ asymptotically; however, steady state is really an asymptotic concept which can not be achieved in a finite simulation run. Thus, because the initial observations are not representative of the steady state distribution and because we have a finite sampling budget, estimators such as the sample mean will be biased. This bias is termed **initialization bias**. Since we can not attain an infinite simulation run, for the purposes of this paper, we consider **steady-state** to occur when the simulation output sequence appears to be covariance stationary. As defined in Law & Kelton (1991, pg. 280), a discrete time stochastic process is said to be covariance stationary if the process has the following properties for $i = 1, 2, \ldots$:

$$E[Y_i] = \theta \quad -\infty < \theta < \infty$$
$$\mathrm{Var}[Y_i] = \sigma^2 \qquad \sigma^2 < \infty$$

and $\mathrm{Cov}(Y_i, Y_{i+j})$ is independent of i for $j = 1, 2, \ldots$.

One way to negate the effects of the initial conditions is to use an extremely long run length to overwhelm the initial "bad" data. The output process for the waiting times in a M/M/1 queue started from the empty and idle state is a good example of a process with a lengthy transient period which requires long simulation run length. As Whitt (1989) points out, we would have to run a M/M/1 queueing model with a traffic intensity of 0.90, for 13,500,00 observations to obtain a parameter estimate with 1% absolute error with respect to the true expected value. This takes approximately 25 hours of real time on a Quadra 840AV using Extend^{TM}. Clearly, long run lengths translate into unacceptable amounts of real computing time.

Two other basic approaches have been used to alleviate the effects of initialization bias. The first approach is to try to set the initial conditions of the simulation to conditions which are more representative of steady state. We can set the initial conditions of a system either deterministically or stochastically. Kelton (1989) explores both methods for replication deletion and batch means. Deterministically setting the initial conditions involves setting the initial state of the queueing system to a constant value such as the mode of the steady state distribution and running either batch means or replication deletion with this initial condition. The problem with setting the initial conditions deterministically is that it ignores the other possible states of the system. Stochastically setting the initial conditions involves sampling from the steady state distribution to obtain the initial state. The problem of course with setting initial conditions is that we do not know the steady state distribution. Kelton demonstrates how using geometric and uniform distributions as approximations for the steady state distribution of the number of customers in the system can help to alleviate initialization bias. Whitt (1993) develops approximations for the steady state distributions of the number in the system. We propose using these approximations to stochastically set the initial conditions.

The second approach is to truncate an initial portion of the sample with the remaining data being used to provide an estimate of the desired performance measure. The basic challenge is to determine a truncation point which controls the initialization bias without increasing the variance of the estimator. Several researchers have developed methods to truncate initial transient observations. Heuristics by Schruben (1982), Fishman (1972), Welch (1983), Conway (1963), Gafarian (1978), Law and Kelton (1983), White and Minnox (1994) and others differ in their approach to controlling the initialization bias, but all incorporate some form of truncation. Schruben's work is unique in that it allows for the testing for the presence of initialization bias. Most truncation heuristics require pilot runs to determine the characteristics of the output data and an estimate of the truncation point. The Welch plot allows us to visualize the transient period across pilot runs and obtain a "guesstimate" of the average warm-up period, based upon these pilot runs. The sample paths averaged across pilot runs gives the simulator a visual clue of where the covariance stationary phase begins. At that point, the simulator would clear the statistics (or truncate) and begin a new estimate. We must not lose sight of the fact that this is a plot across replications so it is merely an estimate of the average warm-up period. Pilot run heuristics do allow us to gain insight to the system and then use this knowledge to run our experiments. Nevertheless, the waste of computational budget time is a shortcoming of pilot run based methods.

The premise behind our methodology is based on considering what the simulator actually gains by performing pilot runs. We argue that the simulator gains knowledge about the steady state behavior of the system which allows for the picking of truncation point. As pointed out in the previous paragraph, this knowledge is really sampled knowledge and may fool the simulator into picking a bad truncation point. We ask the basic question: Why not use knowledge from other sources, e. g. appropriate analytical models, to gain insight into the location of the truncation point? It is the knowledge of where the "leveling" might occur which allows for the picking of the truncation point.

Fishman (1972) discusses the penalty of increased sample variance which may incur from truncation, even though we may see a decrease in the initialization bias. He recommends using the mean squared error(MSE) as an examination of this truncation penalty. The mean squared error is defined by

$$\begin{aligned} \text{MSE} &= E\left[\left(\hat{\theta}-\theta\right)^2\right] \\ &= \text{Bias}^2\left[\hat{\theta}\right] + \text{Var}\left[\hat{\theta}\right] \end{aligned}$$

where $\hat{\theta}$ is an estimator of θ. We consider incorporating knowledge of where the "leveling" might occur by using the approximations in Whitt (1993) as the true mean in a heuristic based upon the MSE criteria.

The rest of this paper is structured as follows. First, we define what we mean by obtaining a better estimate and then cover our basic truncation procedures. We then show the sensitivity of the truncation heuristic to the error of the approximation and give an example comparison of the performance of the truncation heuristics versus the untruncated sample mean. Finally, we summarize our conclusions and give some ideas for future research.

2 METHODOLOGY

In this section, we explain the basic underlying principles of our heuristic in conjunction with the replication/deletion approach to steady state simulation. For explanations and results in the area of batch means, we refer the reader to Delaney (1995). Our methodology differs from other truncation heuristics, in that we assume no knowledge of the system based upon pilot runs. Whitt shows that derived approximations for the long-run distribution of the number in a system and in a GI/G/m queue are quite accurate with the worst absolute error being about 20%. We show that with an approximation of even 20% absolute error relative to the true value we can obtain a "better" estimate than the simulation run mean. We define the term "better" to be an estimate that is both precise(less variable) and accurate(less biased).

The mean squared error criteria considers both the bias and the variance of the estimator. An estimate of the mean squared error of the estimator would be as follows:

$$\widehat{\text{MSE}}\left[\hat{\theta}\right] = \widehat{\text{Bias}}^2\left[\hat{\theta}\right] + \widehat{\text{Var}}\left[\hat{\theta}\right] \tag{1}$$

Assuming that the true value is known and that the Y_i are independent, the bias and the variance of the truncation estimator could be estimated by

$$\widehat{Bias}\left[\hat{\theta}\right] = \bar{Y}(n,d) - \theta \tag{2}$$

$$\widehat{\text{Var}}\left[\hat{\theta}\right] = \frac{1}{(n-d)^2}\sum_{i=d+1}^{n}(Y_i-\theta)^2 \tag{3}$$

where n and d represent the total sample size and the amount of initial data to be deleted and

$$\bar{Y}(n,d) = \frac{\sum_{i=d+1}^{n} Y_i}{n-d}$$

is the sample mean. Clearly, we do not know the true value and the data within a simulation run will almost surely not be independent. We handle the first problem by using an approximation. The problem of dependent data is more difficult. We concede that more appropriate ways for estimating the variance of the estimator exist, e. g. batch means, standardized time series, etc. but we argue for heuristic simplicity. In any event, better methods to estimate the variance of the estimator can easily be incorporated into our methodology and should be considered as future research topics.

Let θ_a be an approximation for θ. This approximation may come from any source, but for our purposes we assume that it is available from a corresponding analytical model. Our first heuristic is based on replacing θ with θ_a in Equations (2) and (3) to obtain an approximate estimate of MSE from Equation (1) and then finding the value of d which minimizes the resultant estimate for the mean squared error. Let us use, $\widehat{\text{MSE}}_a(n,d)$ to indicate Equation (1) when θ_a has been substituted for θ and to indicate that it is dependent on both n and d. The truncation point, d^* is determined by

$$d^* = \underset{0 \le d < n}{\arg\min} \quad \widehat{\text{MSE}}_a(n,d)$$

Using $\widehat{\text{MSE}}_a(n,d)$ and the corresponding d^* as a truncation point allows us to do the best we can from a finite output sequence in terms of finding both a less biased and less variable estimate. We call this truncation heuristic, MSEAT, for mean squared error approximation truncation. The behavior of $\widehat{\text{MSE}}_a(n,d)$ is illustrated in Figure 1. The sample path is based on $n = 10000$ and has a unique behavior with a truncation point of $d^* = 913$. We examined the behavior of $\widehat{\text{MSE}}_a(n,d)$ over a variety of sample paths for this case. Other values of d^* included $d^* = 1999$ and $d^* = 9706$. Based on this behavior, it should be clear that it may be necessary to truncate a significant portion of the sample.

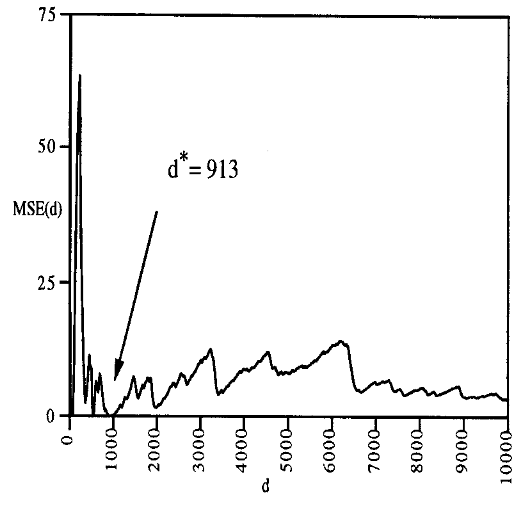

Figure 1: Illustration of $\widehat{\text{MSE}}_a(n,d)$ Behavior

Another way to think of this heuristic is to consider the procedure as picking a subsequence of $\bar{Y}(n,d)$. In other words, for each value of $d = 0, 1, 2, \ldots, n-1$, we have a shorter subsequence from the cumulative sample mean. We are picking the subsequence which is closest to the true mean according to the mean squared error criteria. This causes subsequences which are close to the true mean to be more likely to be chosen; however, because we are replacing θ with θ_a we are picking a subsequence which is close to the approximation. If the approximation has significant error with respect to the true mean, then the estimate of the performance parameter due to truncation may be worse than the untruncated sample mean.

To test this concern, we ran 41 experiments of 30 replications with 10,000 customers exiting the system of an M/M/1 queue with common random numbers. The performance measure of interest is the steady state expected waiting time in the queue. We used the true analytical value as the initial "approximation" and then perturbed the truth incrementally by 0.01 absolute error to give us an absolute error range from 0 to 20%. We note that Whitt's approximations reduce to the exact values for the M/M/1 case. Figure 2 depicts the findings across the 41 experiments. In the figure, the upward sloping lines represent the performance of the heuristic for positively perturbed values of θ_a while the downward sloping lines represent performance for negatively perturbed θ_a. The bands represent 95% confidence intervals across the 30 replications. The final values of 16.34 and 15.97 shown are for the case of 20% error. The average for the untruncated sample mean across the 30 replications was 14.85 with a 95% confidence confidence interval of $[14.75, 14.95]$ indicating clear negative initialization bias. The significant result was that even with an approximation absolute error of 20% the truncation estimate was less biased than the run mean with no truncation. The fact that the average across 30 sample paths could not produce an estimate within the 95% confidence band of the negatively perturbed approximation heuristic is even more significant. We

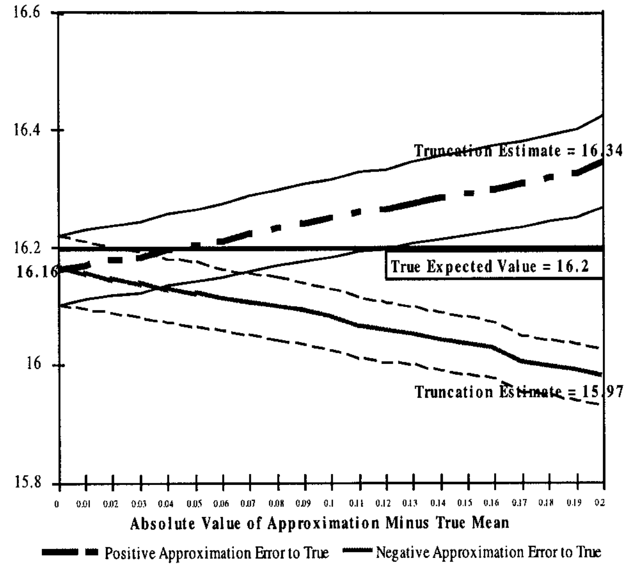

Figure 2: Approximation Sensitivity

did note that the strength of the approximation in the bias and variance equations forced the resultant estimate towards the approximation. We considered reducing the strength of our heuristic to let the sample data have a more determinant impact on the estimate. We felt that maintaining the bias equation and using a different equation for the variance would lessen the strength of the heuristic, but still allow for reduced variance and less bias.

White and Minnox (1994) describe a truncation heuristic based on finding d such that the sample half-width is minimized. This equates to finding the value of d which minimizes the sample standard deviation. While this indeed resulted in an improved estimator, we found that using it to determine an estimate of the expected wait time in the queue occasionally resulted in extremely biased estimates when the sample paths had small and similar observations at the tail end of the simulation run. In some sense, it relies too heavily on sample path information while our basic heuristic relies too heavily on the approximation. Thus, we felt that the following heuristic was appropriate.

$$d^* = \underset{0 \le d < n}{\arg\min} \quad \widehat{\text{Bias}}^2\left[\hat{\theta}\right] + \widehat{\text{Var}}\left[\hat{\theta}\right]$$

where

$$\begin{aligned} \widehat{Bias}\left[\hat{\theta}\right] &= \bar{Y}(n,d) - \theta_a \\ \widehat{\text{Var}}\left[\hat{\theta}\right] &= \frac{\sum_{i=d+1}^{n}\left(Y_i - \bar{Y}(n,d)\right)^2}{(n-d)(n-d-1)} \end{aligned}$$

We refer to this heuristic as, MSEASVT, for mean squared error approximation and sample variance truncation. We used the algorithm given in Figure 3 to apply MSEASVT during replication/deletion. We find an optimal d for each replication and its corresponding truncated sample mean. The final point estimator is the average across replications of truncated sample means. Variations of our basic heuristic are also possible. For example, one might apply the algorithm over a range of values of θ_a. For example, apply the MSEASVT heuristic for three values $\theta_a - \epsilon, \theta_a, \theta_a + \epsilon$ and average the resultant point estimators. We refer the interested reader to Delaney (1995) for further evaluation of these types of heuristics. Our basic replication/deletion algorithm requires that the data from within a replication be saved and then analyzed at the end of a replication. For adequately long run lengths this can be a significant amount of data. As suggested by an anonymous reviewer, if one could store *all* of the data from *each* replication then many other variations are also possible. For example, one might find d^* for each replication and use the average of the d^*'s for the truncation point for every replication. In addition, one could compute the average MSE(d) curve across the replications and then find the minimum d for the average curve. These possibilities remain to be explored.

3 SIMULATION PROGRAM DESCRIPTION

The simulations were performed using the Extend simulation package. The package allows the simulator to use established blocks for discrete and continuous simulations using drag and drop technology. By simply connecting blocks, a simulation can be ready to run within 5 minutes of your first introduction to the package; however, as with most simulation packages of this nature more complex modeling and control

1. Initialize: r; n;
2. Compute Approximation: θ_a;
3. Run Replications:

```
    ysum ← 0; minmse ← ∞;
    for j ← 1 to r
      initialize simulation
      sum ← 0; sumsq ← 0;
      for i ← 1 to n
        simulate y[i];
        sum ← sum + y[i];
        sumsq ← sumsq + y[i] * y[i];
      endfor
      for d ← 0 to n − 2
        avg ← sum/(n − d);
        bias ← avg − θa;
        var ← ((sumsq − sum * sum/(n − d));
        var ← var/((n − d) * (n − d − 1));
        mse ← bias * bias + var;
        if mse ≤ minmse
          minmse ← mse; ybar ← avg;
        sum ← sum − y[d + 1];
        sumsq ← sumsq − y[d + 1] * y[d + 1];
      endfor
      ysum ← ysum + ybar;
    endfor
```

4. $\hat{\theta} \leftarrow ysum/r$;

Figure 3: MSEASVT Algorithm

of the simulation requires the simulator to modify existing blocks to create custom blocks to be used in the solution process. This is made easier in Extend since the underlying language is ModL which is a variant of C. Dialog boxes for user input are also easy to create. For simulating the GI/G/m queue, it is simply a matter of utilizing an arrival generator block, a queue block, and a resource block. Other blocks were created to stochastically set the initial state of the GI/G/m simulations based upon Whitt's approximations, to control the random number seeds, and to apply the truncation heuristics at the end of a simulation run. In comparing empty and idle initial conditions to stochastically set initial conditions, we utilized common random numbers. Whitt's approximation for the expected waiting time in the GI/G/m queue is based on a system approximation approach which modifies the result of the standard M/M/m queue. The basic functional relationship has the following form

$$E\left[W_{q_{GI/G/m}}\right] = \Phi(\rho, c_a^2, c_s^2, m) E\left[W_{q_{M/M/m}}\right]$$

where ρ is the traffic intensity, c_a^2 is the coefficient of variation for the inter-arrival distribution, c_s^2 is the coefficient of variation for the service distribution, and m is the number of servers. The function $\Phi(\rho, c_a^2, c_s^2, m)$ can be thought of as a correction factor and has a complex form, but is still easy to compute. Whitt also provides approximations for the steady state queue length and number in the system distributions. Whitt uses the geometric distribution as the building block for his queue length distribution. The reader is referred to Whitt (1993) for more on the development of these approximations. Algorithms for implementing Whitt's approximations and the truncation heuristics are given in Delaney (1995).

4 SIMULATION EXPERIMENTS AND RESULTS

To test our heuristics, we set up a series of non-standard queueing models with a traffic intensity $\rho \geq 0.9$. We used models other than the M/M/c model since Whitt's approximations are exact for that case. To check our results we used Hillier and Yu (1981) and Tijms, Seelen and Van Hoorn (1985). We analyzed our models by stochastically setting the initial conditions and then, with recorded seeds, empty and idle. We performed this analysis for batch means and replication deletion. Table 1 is a representative sample of our experiments for the replication deletion approach. For further experimental results including the batch mean comparisons, we refer the reader to Delaney (1995).

Table 1: Table of Experiments

ρ = traffic intensity, IC = initial conditions
EI = empty and idle, SS = stochastically set
r = # micro-replications
R = # macro-replications
n = micro-replication run length

Exp	Model	ρ	IC	r	n	R
1	$E_2/E_2/4$	0.9	SS	30	10,000	25
2	$E_2/E_2/4$	0.9	EI	30	10,000	25
3	$U/LN/3$	0.9	SS	10	30,000	25
4	$U/LN/3$	0.9	EI	10	30,000	25
5	$E_2/E_2/4$	0.98	SS	10	21,000	25
6	$E_2/E_2/4$	0.98	EI	10	21,000	25

Table 2 reports the average absolute value of the bias over the R = 25 macro-replications. In all of the cases, the average absolute value of the bias for the truncation heuristics is lower than the run mean's average absolute value of the bias. On average across the experiments, 83% of the time there was a bias reduction. For the truncation heuristics roughly 46% of the data is being truncated. Table 3 reports the average half-width across the 25 macro-replications. On average across the experiments, 97% of the time there was a reduction in the half-width for the truncation heuristics as compared to the run mean's half-width. The specified coverage probability was 95%. The average estimated coverage across the experiments for the run mean was 86.7% while the average across the truncation heuristics was 63.2%. Some truncation heuristics had extremely bad coverage, but this is due to the extremely small half-width obtained. We might note that at times when the truncation heuristics increased the bias they did so as compared to the run mean. This was primarily due to the heuristics tendency to pick a subsequence which is close to the approximation. This does not mean that the heuristics gave a bad estimate, only that for that macro-replication the run mean was closer to the true mean value. In fact, as shown in Figure 4, the truncation heuristic was within 2% of the true value significantly more often than the untruncated sample mean.

Table 2: Bias Macro-Results

$\bar{b}$ = average absolute bias
hw = half-width for 95% c.i. on $\bar{b}$
%dt = % data truncated
ntbr = # times bias reduced given $R = 25$

Exp	Method	$\bar{b} \pm hw$	ntbr	%dt
1	Run Mean	0.27 ± 0.10	-	0.00
	MSEAT	0.11 ± 0.03	18	60.82
	MSEASVT	0.13 ± 0.04	18	44.97
2	Run Mean	0.26 ± 0.08	-	0.00
	MSEAT	0.10 ± 0.03	20	61.52
	MSEASVT	0.10 ± 0.03	20	45.25
3	Run Mean	0.08 ± 0.02	-	0.00
	MSEAT	0.02 ± 0.004	24	61.53
	MSEASVT	0.02 ± 0.004	24	44.27
4	Run Mean	0.07 ± 0.02	-	0.00
	MSEAT	0.03 ± 0.004	22	60.52
	MSEASVT	0.02 ± 0.004	22	41.52
5	Run Mean	11.27 ± 3.25	-	0.00
	MSEAT	4.09 ± 1.40	20	60.12
	MSEASVT	4.12 ± 1.39	20	57.66
6	Run Mean	12.05 ± 3.20	-	0.00
	MSEAT	4.56 ± 1.49	20	63.00
	MSEASVT	4.55 ± 1.48	20	59.00

The mean squared error criteria punishes those subsequences which are far from the approximation and conversely rewards those close to the approximation. A different criteria such as the mean absolute

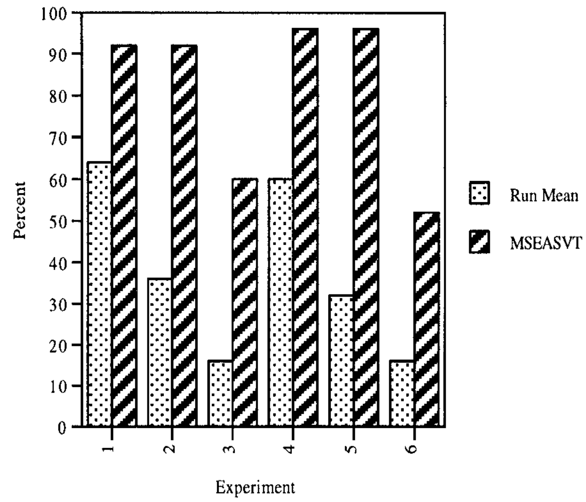

Figure 4: Percent Within 2% of True

deviation might be more appropriate. In any case, on average the performance of the truncation heuristics was superior to using the untruncated run mean.

5 CONCLUSIONS

We can see from the results that using queueing approximations to stochastically set the initial conditions of the system reduces the initialization bias of a performance parameter estimate when we have a finite computer budget. Although a finite run length may not get us through a transient period, our results lead us to believe that stochastically setting the initial conditions will reduce the size of the transient period. Additionally, using the approximations to perform a back-end truncation of output data at the point where the minimum estimated MSE occurs reduces the bias of the estimate. Even if an approximation had an absolute error of 20% with respect to the true value, it could still produce a point estimate closer to the true expected value when compared to the untruncated sample mean. Though the coverage across the 25 experiments for each model was not what we desired, one can easily see that the confidence half-width across each experiment was significantly smaller than that of the untruncated run mean data. This improvement in the precision is the reason for the shortage of coverage. We consistently produced estimates closer to the true value of the mean.

While we do acknowledge that we should continue our research with more complex tests on more complicated queueing systems, the preliminary findings of stochastic initialization support the results given in Kelton (1989). Kelton (1989) found that the approximation assisted point estimate heuristics with no pilot runs lessen the possibility of incorrect inference from output data. Indeed, the $E_2/E_2/4$ model with $\rho = 0.98$ demonstrates that if you have an expected queue length which is large, using Whitt's approximations for the steady state distributions allows us to achieve a significantly better estimate than an empty and idle system. Using our heuristic truncation rule then allows us to get a less biased and less variable estimate due to the large transient phase for this case. We should also continue our evaluation on more complex queueing systems such as networks of queues. We note that software packages already exist, e. g. MANUPLAN & SIMSTARTER, see Suri et. al. (1990), which will translate an analytical queueing analysis into a corresponding simulation program. The queueing analysis could then be used to improve the simulation estimation process.

Table 3: Half-Width and Coverage Macro-Results

$\bar{hw}$ = average half-width
hw = half-width for 95% confidence interval on $\bar{hw}$
%c = estimated % coverage
nhwr = # half-width reduced given $R = 25$

Exp	Method	$\bar{hw} \pm hw$	nhwr	%c
1	Run Mean	0.80 ± 0.06	-	96.0
	MSEAT	0.21 ± 0.05	25	80.0
	MSEASVT	0.22 ± 0.05	25	80.0
2	Run Mean	0.80 ± 0.06	-	96.0
	MSEAT	0.18 ± 0.04	25	60.0
	MSEASVT	0.20 ± 0.04	25	60.0
3	Run Mean	0.12 ± 0.01	-	88.0
	MSEAT	0.01 ± 0.008	25	20.0
	MSEASVT	0.02 ± 0.008	25	32.0
4	Run Mean	0.12 ± 0.01	-	92.0
	MSEAT	0.01 ± 0.008	25	20.0
	MSEASVT	0.02 ± 0.008	23	32.0
5	Run Mean	19.91 ± 2.73	-	76.0
	MSEAT	5.85 ± 1.06	23	80.0
	MSEASVT	5.80 ± 1.06	23	80.0
6	Run Mean	19.32 ± 2.62	-	72.0
	MSEAT	6.33 ± 1.04	24	68.0
	MSEASVT	6.23 ± 1.03	24	72.0

The truncation methodologies do lend themselves to a batch means analysis of one long run; however, the stochastically set initial condition point of the one run negates the power of the steady state distribution approximation across multiple sample paths. The truncation heuristic could be applied to the entire data set in which case there is the potential for a significant amount of data to be deleted. Applying batch means to this reduced sample is problematic. Alternatively, the simulator can reserve part of the data and only apply the truncation heuristic to the first part of the data set. For example, the simulator could apply the heuristic to the first γ% of the data and batch the other part of the sample. For more details of this, we refer the reader to Delaney (1995).

Our methods meet our stated objective of obtaining a "better" estimate of the expected wait time in the queue. We do not satisfy our implied goal of 95% coverage across our experiments. We realize, however, that this is directly attributable to the exact precision that the heuristics impart. Indeed, the half-widths produced by replication deletion in combination with the heuristics incorporating the approximations was considerably smaller than that with no truncation. What resulted was a series of estimates each with an average bias significantly less than the run means *and* improved precision. This is a very intriguing result since it implies that there may be a truncated subsequence which has very good statistical properties. Future research should explore exploiting this method as a possible variance reduction technique. The question still remains as to which is better; better coverage with the possibility of significant bias and variance or less coverage with little or no bias and reduced variance? The decision maker is the only one who can answer this fundamental question. We submit, however, that since statistical analysis and inferences of output data can lead to errors based solely on the data, using the less biased and less variable estimate should ultimately result in better decision outcomes.

REFERENCES

Conway, R.W. 1963. Some Tactical Problems in Digital Simulation. *Management Science* Vol. 10, No. 1, pp. 47-61.

Delaney, P. 1995. Control of Initialization Bias in Queueing Simulations Using Queueing Approximations. unpublished M.S. Thesis, Department of Systems Engineering, University of Virginia.

Fishman, George S. 1972. Bias Considerations in Simulation Experiments. *Operations Research* Vol. 20, pp. 785-790.

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.

Hillier, Frederick S. and Oliver S. Yu. 1981. *Queueing Tables and Graphs*, New York: North Holland.

Kelton, W. David. 1989. Random Initialization Methods In Simulation. *IIE Transactions* Vol. 21, pp. 355-367.

Law, Averill M. and W. David Kelton. 1991. *Simulation Modeling & Analysis*, 2d Edition, USA: McGraw-Hill, Inc.

Law, Averill M. and W. David Kelton. 1983. A New Approach for Dealing with the Startup Problem in Discrete Event Simulation. *Naval Research Logistics Quarterly* Vol. 30, pp. 641-658.

Schruben, Lee W. 1982. Detecting Initialization Bias in Simulation Output. *Operations Research* Vol. 30, No. 3, pp. 569-590.

Suri, R., D. Derleth, and M. Tomsicek. 1990. 'Modeling Manufacturing Systems Using MANUPLAN and SIMSTARTER - A Tutorial. In *Proceedings of the 1990 Winter Simulation Conference*, ed. Osman Balci, Randall Sadowski, and Richard E. Nance, pp. 168-176.

Tijms, Henk C., L.P. Seelen and M.H. VanHorn. 1985. *Tables for Multi-Server Queues*, North Holland: Elsevier Science Publishers.

Welch, Peter D. 1983. A Graphical Approach to the Initial Transient Problem In Steady State Simulations. In *10th IMACS World Congress on System Simulation and Scientific Computation*, pp. 219-221.

White, K. Preston and Mary A. Minnox. A Simple Rule for Mitigating Initialization Bias in Simulation Output. To appear in *Proceedings of the 1994 IEEE Conference on Man, Systems and Cybernetics*.

Whitt, Ward. 1989. Planning Queueing Simulations. *Management Science* Vol. 35, No. 11, pp. 1341-1366

Whitt, Ward, 1993. Approximations for the GI/G/m Queue. *Production and Operations Management* Vol. 2, No. 2, pp. 114-161.

Wilson, James R. and Alan B. Pritsker. 1978a. A Survey of Research on the Simulation Startup Problem. *Simulation*, Sept., pp. 55-58.

Wilson, James R. and Alan B. Pritsker. 1978b. Evaluation of Startup Policies in Simulation Experiments. *Simulation* , Sept., pp. 79-89.

AUTHOR BIOGRAPHIES

MANUEL D. ROSSETTI is an Assistant Professor in the Systems Engineering Department at the University of Virginia. He received his Ph.D. in Industrial & Systems Engineering from The Ohio State University. His research interests include the design, analysis, and optimization of manufacturing, telecommunication, and transportation systems using stochastic modeling, computer simulation, and artificial intelligence techniques. Dr. Rossetti is an Associate member of the Institute of Industrial Engineers and a member of the IIE OR Division. Dr. Rossetti is also a member of INFORMS and SCS.

CPT PATRICK J. DELANEY is an Instructor within the Systems Engineering Department at the United States Military Academy at West Point. He received his Master of Science Degree in Systems Engineering from the University of Virginia in May 1995. His research interests include simulation modeling and analysis for decision making within the U.S. Army.