EFFICIENT TRUNCATION POINT ESTIMATION FOR ARBITRARY PERFORMANCE MEASURES

Mirko Eickhoff, Donald C. McNickle, Krzysztof Pawlikowski
University of Canterbury
Christchurch, New Zealand
E-Mail: m.eickhoff@cosc.canterbury.ac.nz

ABSTRACT

In steady-state simulation the output data of the transient phase often causes a bias in the estimation of steady-state performance measures. Common advice is to cut off this transient phase. Finding an appropriate truncation point is a well-known problem and is still not completely solved. In this paper we propose an improved algorithm for the determination of the truncation point. The required run time of this algorithm is substantially improved without reducing the reliability of the results. Because this algorithm is based on comparisons of empirical cumulative distribution functions, the truncation point is valid for any arbitrary steady-state performance measure, such as the mean, variance or quantiles.

INTRODUCTION

In discrete-event simulation the sequence of output data is a stochastic process $\{X_j\}_{j=1}^{\infty}$. The observations of this process are usually correlated and influenced by the initial state I of the system chosen by the analyst. Let $F_j(x|I) := \Pr[X_j \leq x|I]$ denote the cumulative distribution function of X_j. Assuming an ergodic system $F_j(x|I)$ is converging towards $F(x) = \lim_{j\to\infty} F_j(x|I)$ which is called the marginal cumulative distribution function of the process $\{X_j\}_{j=1}^{\infty}$ in steady-state. The primary concern of steady-state simulation is to determine this distribution or derived measures.

In general the influence of I is significant in the beginning and decreases with increasing model time. If the interest is focused on the steady-state behavior of the system, this initialization bias is obviously undesirable. A common way to reduce the influence of I is to truncate the "most" influenced part of the stochastic output process $X_1, \ldots, X_{l-1}$. Following this strategy the problem is to find an appropriate truncation point l. In the literature the steady-state phase $\{X_j\}_{j=l}^{\infty}$ is, e.g., described as a phase which is "relatively free of the influence of initial conditions" [7] or by the statement that $X_l, X_{l+1}, \ldots$ "will have approximately the same distribution" [11]. In practise there will often be an observation index l, such that

$$\forall j \geq l : F_j(x|I) \approx F(x) \tag{1}$$

is valid. Of course l should be finite, and should be the minimum of all indices, for which equation (1) holds. Even though the estimation of $F(x)$ is the ultimate goal of steady-state simulation, the expected value of the steady-state random variable $\mu = E[X]$ is often the only measure of interest. In this situation it is a generally accepted approach to replace equation (1) by:

$$\forall j \geq l : E[X_j] \approx E[X] \tag{2}$$

In [12] it is shown that in the case of a $M/M/1/\infty$ system, the use of equation (2) works even in quantile estimation. In general, however, the convergence of the mean is only a necessary condition for stationarity, and not a sufficient preliminary (see [15]). Therefore, equation (1) can be applied in analysis of mean, variance, quantiles, or even rare-event estimation and equation (2) should only be used in mean value analysis. In [9] experiments with methods based on equation (2) are performed, and it could be shown that a least one of them does not work at all well. However, the realization of equation (1) is not straightforward. It is therefore not very surprising, that the most common methods for detection of the truncation point are based on a visual inspection of the output data (see e.g. [15]). Other, completely algorithmic methods only give proper results under special conditions (see [13] and [8]) and are mostly based on one long simulation run. Nearly all methods use equation (2), although in the most frequent case the steady-state phase is defined by the convergence of $F_j(x|I)$ towards the steady-state distribution $F(x)$.

In the MRIP scenario (multiple replications in parallel, see [14]) it is possible to observe a random sample of p (number of replications) independent and identical distributed realizations of each X_j. Let x_{ij} be the jth observation of the ith replication, with $1 \leq i \leq p$ and $1 \leq j < \infty$. Therefore, the empirical (marginal) cumulative distribution function $\tilde{F}_j(x|I)$ based on the order statistic of $x_{1j}, \ldots, x_{pj}$ is an estimate of $F_j(x|I)$. The use of independent replications is analyzed for the situation of mean value estimation in [16] and [1]. The main source of error using independent replications is the initialization bias. If this source of error is eliminated, the estimates of independent replications are more accurate than the ones of e.g. batching methods. In [6] an approach is proposed which is based on equation (1) and uses the MRIP scenario. Its performance is examined and

improved in [3] and [4]. An application of this approach is demonstrated in [2]. The results show that this approach is much more reliable than methods which are based on equation (2), especially if the transient behaviour is more complicated. Unfortunately, however, the computing time of this approach is quite large, possibly too excessive, and in applications the computing time is as important as reliability.

In the next sections an algorithm is described, which is based on equation (1). In comparison to the algorithm described in [4], the required run time of the new algorithm is substantially improved without reducing the reliability of the results. This is examined in a later section by comparing the worst-case run time and the reliability of both algorithms. In this paper only sequential methods and algorithms which are based on a dynamic set of data are considered. Therefore, the output analysis is performed online and guides the simulation experiment until estimates are statistically reliable.

IMPROVED ALGORITHM

Listing 1 shows the improved algorithm in pseudo code, where for convenience some special notations are used. Let the sequence $\{y_{ij}\}_{i=1}^{p}$ be the order statistic of $\{x_{ij}\}_{i=1}^{p}$. Using the operators +, -, / and := in conjunction with sequences (see lines 0, 5, 8, 10 and 15) means to use these operators on each component separately. The operator $\approx$ in line 10 and 15 implements equation (1) and is realized by the Kolmogorov-Smirnov two-sample test (KS-test). If the null hypothesis of equality is accepted the operator $\approx$ (resp. the KS-test) results *true*. The method *observe()* collects one observation of each replication and the method *uniform(a,b)* delivers a uniform distributed integer random number between *a* and *b* used as index. The variable l is the actual candidate for the truncation point, at the end of the algorithm it is the valid truncation point. The variable n is the number of observations collected of each replication so far. r is the ratio between the transient phase and the observed part of the steady-state phase (see [3]).

Listing 1: Pseudo code of the improved algorithm

```
0  int l := 0; int n := 0; int r := 10; {s_i}_{i=1}^p := 0;
   bool NoTestFailed := false;
   while (¬NoTestFailed){
      n := n + 1;
      observe({x_in}_{i=1}^p);
5     {s_i}_{i=1}^p := {s_i}_{i=1}^p + {y_in}_{i=1}^p;
      if (0 ≠ n mod (r + 1)) continue;
      l := l + 1;
      {s_i}_{i=1}^p := {s_i}_{i=1}^p − {y_il}_{i=1}^p;
      NoTestFailed := true;
10    if (¬({y_il}_{i=1}^p ≈ {s_i}_{i=1}^p/(n − l)))
         NoTestFailed := false;
      for (int k := 1; k ≤ r; k := k + 1){
         if (¬NoTestFailed) break;
         int u := uniform(lk + 1, l(k + 1));
15       if (¬({y_il}_{i=1}^p ≈ {y_iu}_{i=1}^p))
            NoTestFailed := false;
      }
   }
```

The most time consuming factor of the algorithm described in [4] is the increasing amount of KS-tests during each step of the algorithm. The improved algorithm described in Listing 1 avoids these tests by using the calculated sequence $\{s_i\}_{i=1}^{p}$ which is an estimate of $F(x)$ based on the latest observations during each step of the algorithm. The sequence $\{s_i\}_{i=1}^{p}$ is the sum of all order statistics which are not part of the transient period. New observations are added whereas observations of the transient period are subtracted from $\{s_i\}_{i=1}^{p}$ (see lines 5 and 8). Dividing each component of $\{s_i\}_{i=1}^{p}$ by the number of addends results in an estimate of $F(x)$. This sequence is compared with the order statistic of the actual test sample $\{x_{il}\}_{i=1}^{p}$ (see line 10). Because $\{s_i\}_{i=1}^{p}$ is calculated of observations at different model times, a possible periodic behaviour could be overlooked and an unreliable estimate of the truncation point would be possible (cf. [4]). Therefore additional r randomly chosen sequences are used to avoid this trap. The observed part of the steady-state phase is divided into r equally sized intervals. Each interval contains one randomly chosen sequence. In a loop all of these sequences are compared with the actual test sample $\{x_{il}\}_{i=1}^{p}$ (see lines 12 to 17). If the assumption of equality is rejected by the KS-test for $\{s_i\}_{i=1}^{p}$ or any of the randomly chosen test samples the truncation point l is not adequate and more steps of the algorithm have to be performed. In contrary to the previous version of this algorithm, the number of needed KS-tests in each step is constantly $r + 1$.

WORST CASE TIME COMPLEXITY

In [4] it is demonstrated by experiments with a number of different kinds of transient behaviour that the previous version of the algorithm described in Listing 1 is very reliable and has therefore a great advantage over some other commonly used methods for truncation point estimation. But the price for this reliability is the high requirement of run time which is $O(n^2 p \log(p))$. This run time is possibly too exhaustive for efficient implementations in application. In this section it will be shown that the algorithm described in the previous section is a substantial improvement, because its run time is only $O(np \log(p))$ without significant loss of reliability in conjunction with any kind of transient behaviour.

As before, let p denote the number of replications and n the amount of observations of each single replication. Note, that especially at the end of the algorithm n is a multiple of $r + 1$. The total number of observations is pn. Assume, that all basic arithmetic operations are in $O(1)$ (cf. [5]). In the following the mentioned running times consider worst-case time complexity.

Theorem
The worst case running time of the algorithm described in Listing 1 is $O(np \log(p))$.

Proof: Only KS-tests with random samples of size p are performed. The basis of the KS-test are two sorted random samples. Sorting can be done in $O(p \log(p))$. To de-

termine the maximum difference of the compared samples a pointer has to be shifted through each sorted random sample. This can be done in $2p$ steps which leads to $O(p)$. To accept or reject the null hypothesis the determined maximum difference has to be compared with a tabulated critical value. This can be done in $O(1)$. Therefore, the run time of one KS-test is $O(p\log(p)) + O(p) + O(1) = O(p\log(p))$.

The run time of a single execution of lines 3 and 6 is $O(1)$ and of lines 4 and 5 it is $O(p)$. Because the while-loop in line 2 is executed n times before the algorithm stops, the run time of this part of the algorithm is $O(np)$.

A single execution of lines 7, 9 and 11 can be done in $O(1)$; a single execution of line 8 can be done in $O(p)$; to execute line 10 a run time of $O(p\log(p))$ is needed, because a KS-test has to be performed. Because of the condition in line 6 this part of the algorithm is executed only $\frac{n}{r+1}$ times. Therefore, the run time of this part of the algorithm is $\frac{n}{r+1} \cdot O(p\log(p))$ which leads to $O(np\log(p))$ because r is a constant parameter.

A single execution of lines 12, 13, 14 and 16 needs only a minor run time of $O(1)$. The KS-test in line 15 can be done in $O(p\log(p))$. The for-loop is executed at maximum r times, therefore, the run time of one complete for-loop in each step of the algorithm is $r \cdot O(p\log(p))$. All in all $\frac{n}{r+1}$ for-loops have to be performed. Therefore, the run time of this part of the algorithm is $n \cdot \frac{r}{r+1} \cdot O(p\log(p))$ which leads to $O(np\log(p))$.

Combining all results, the run time of the algorithm is $O(np) + O(np\log(p)) + O(np\log(p)) = O(np\log(p))$ ■

Because p could be considered as a constant parameter and usually $p << n$ holds, the run time could be described by $O(n)$. This run time is linear and highly efficient, because each observation has to be processed at least once.

PERFORMANCE

To compare the reliability of both versions of the algorithm the same artificial processes as in [4] are used, except the starting condition of the ARMA process is changed, to create a more unusual initial state. The output data of these artificial processes is used to show the reliability on many different kinds of transient behaviour. Especially for those, where the use of equation (2) leads to unreliable results. Let $\{\epsilon_t\}_{t=1}^{\infty}$ denote an independent Gaussian white noise process (see [10]). In all experiments we used $p = 100$, $r = 10$ and an α-level of 0.05 for the critical value of the KS-test. For details on this parameters the reader is referred to [3].

Process A: linear transient mean

$$Y_t^{(1)} = f_t^{(1)} + \epsilon_t \, , \, f_t^{(1)} = \begin{cases} x - t\frac{x}{l} & \text{if } t < l, \\ 0 & \text{else.} \end{cases}$$

with $x = 10$, $l = 100$

Process B: linear transient variance

$$Y_t^{(2)} = f_t^{(2)} \cdot \epsilon_t \, , \, f_t^{(2)} = \begin{cases} x - t\frac{x-1}{l} & \text{if } t < l, \\ 1 & \text{else} \end{cases}$$

with $x = 10$, $l = 100$

Process C: exponential transient mean

$$Y_t^{(3)} = f_t^{(3)} + \epsilon_t \, , \, f_t^{(3)} = x \cdot e^{(t\frac{\ln(0.05)}{l})}$$

with $x = 10$, $l = 100$

Process D: ARMA$(5, 5)$

$$Y_t^{(4)} = 1 + \epsilon_t + \sum_{i=1}^{5} \frac{1}{2^i}(Y_{t-i} + \epsilon_{t-i})$$

with $Y_{-5}^{(4)} = Y_{-4}^{(4)} = Y_{-3}^{(4)} = Y_{-2}^{(4)} = Y_{-1}^{(4)} = 100$

Process E: periodic

$$Y_t^{(5)} = f_t^{(5)} + \epsilon_t \, , \, f_t^{(5)} = b \cdot \sin(\omega t)$$

with $b = 1$, $T = \frac{2\pi}{\omega} = 50$

Process F: non-ergodic

$$Y_t^{(6)} = f_t^{(6)}\epsilon_t + f_t^{(1)} \, , \, f_t^{(6)} = ct$$

with $x = 10$, $l = 100$, $c = 0.01$

The processes A to D converge towards a steady-state distribution. Therefore equation (1) can be used to estimate the beginning of the steady-state phase. To achieve a solid result of our meta output analysis the simulation experiments with the algorithm described in [4] and the algorithm of Listing 1 are performed 20 times. The average of all results and its confidence interval can be calculated with the standard interval estimator for independent and identically distributed data. The results are listed in table 1.

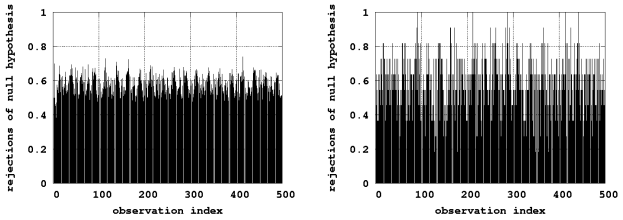
using 20 runs	previous algorithm	improved algorithm
Process A	98.00 ± 0.43 (0.4%)	97.25 ± 0.57 (0.6%)
Process B	86.20 ± 1.01 (1.2%)	82.20 ± 1.87 (2.3%)
Process C	103.05 ± 1.76 (1.7%)	99.25 ± 1.81 (1.8%)
Process D	190.25 ± 6.53 (3.4%)	185.00 ± 6.55 (3.5%)
run time	$O(n^2 p\log(p))$	$O(np\log(p))$

Table 1: This table shows the average estimate of the truncation point and its confidence interval. Each result is based on 20 simulation experiments.

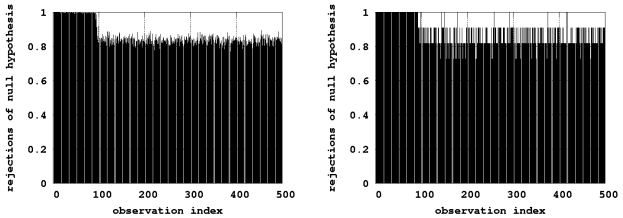
Surprisingly, the halfwidth of the confidence intervals is in all our experiments smaller than five percent of the mean value after fewer than 20 simulation experiments. This shows that both algorithms deliver a robust estimate with a small variance. Even though the results of both algorithms are comparably good, the algorithm described in Listing 1 tends to estimate the truncation point a bit earlier. This shows that the fewer KS-tests of this algorithm are weaker than the large

number of KS-tests in the previous version of the algorithm which cause the long run time.

The processes E and F do not converge towards a steady-state distribution, and so there is no steady-state phase at all. Therefore, both algorithms should not return a truncation point. To check the reliability in this case each step of the algorithms is observed by plotting the number of rejections of the null hypothesis (see Figure 1). The rejections should be on a high level so that equation (1) will not be true for any tested value of l. To make the number of rejections comparable they are standardized by the number of all performed KS-tests. Therefore, the value 1 means that all KS-tests reject the null hypothesis, and the value 0 means that all KS-tests accept the null hypothesis.



(a) Process E / previous algorithm (b) Process E / improved algorithm

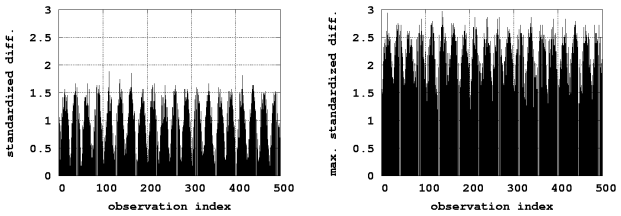


(c) Process F / previous algorithm (d) Process F / improved algorithm

Figure 1: The number of rejections of the null hypothesis standardized by the number of all performed KS-tests. The values are plotted over model time, each peak represents one step of the algorithms.

The plots 1(a) and 1(c) of the algorithm described in [4] are quite smooth, because a large number of KS-tests are used to check equation (1). Additionally these plots show that the KS-tests of the previous algorithm work reliably, because the number of rejected null hypothesis is on a high level when analyzing process E as well as process F. Note, this algorithm would accept an estimate, when the number of rejections is below a certain threshold which is usually set at 0.05. The plots 1(b) and 1(d) of the algorithm described in Listing 1 are not as smooth, because the number of KS-tests is much smaller. However, the number of rejections of the null hypothesis is also on a high level in both cases. Note, the improved algorithm accepts an estimate only, if all KS-test accept the null hypothesis. Therefore the improved algorithm works on process E and F as reliably as the previous algorithm.

Furthermore, process E is a good example to demonstrate that a truncation point estimation exclusively based on $\{s_i\}_{i=1}^{p}$ is not sufficient. In Figure 2 the maximum difference of the actual candidate for the truncation point $\{x_{il}\}_{i=1}^{p}$ is compared with $\{s_i\}_{i=1}^{p}$ (see plot 2(a)) resp. compared with the r randomly chosen samples (see plot 2(b)). In the second case the maximum difference of all r comparisons is plotted. To achieve comparability the maximum difference is standardized by the appropriate critical value of the KS-test. If the standardized maximum difference is below 1 then the null hypothesis would be accepted by the KS-test.



(a) KS-tests with $\{s_i\}_{i=1}^{p}$ (b) KS-tests with r random samples

Figure 2: This Figure shows the maximum differences of the KS-tests. It is distinguished between the KS-tests of lines 10 and 15 (see Listing 1).

Plot 2(a) shows that in periodical distances the KS-test based on $\{s_i\}_{i=1}^{p}$ would accept the null hypothesis. This happens, because $\{s_i\}_{i=1}^{p}$ cannot reflect the periodic behaviour. Observations of different model times are combined which results in the same problem as shown for the method of Welch (see [4] and [15]). However, the KS-tests with the randomly selected samples make sure that the improved algorithm works reliably (see plot 2(b)).

LIMITS

Because equation (1) uses an approximation the estimate of the beginning of the steady-state phase of process A is always smaller than the theoretically best value which is in this example $l = 100$. However, using the equality instead of an approximation does not make sense in output processes such as process C. Using the equality instead of an approximation would lead to an infinite value for l, because in process C the influence of the initial state disappears exponentially.

Figure 3 plots the results of simulation experiments for the process A with different numbers of parallel replications. The experiments are executed as described in the previous section. This plot validates the assertion, that the estimate is always smaller than the theoretically best value. Furthermore, another effect can be observed. The value of the estimate of the truncation point is decreasing with a decreasing number of parallel replications. Note, that the critical values of the KS-test are defined for very small sample sizes, too. The realization of the approximation of equation (1) based on the KS-test gets weaker for smaller numbers of parallel replications. Therefore, we recommend at least 30 parallel replications, and if possible more than 50 parallel replications should be used. This limits are valid for the algorithm described in [4] and in Listing 1.

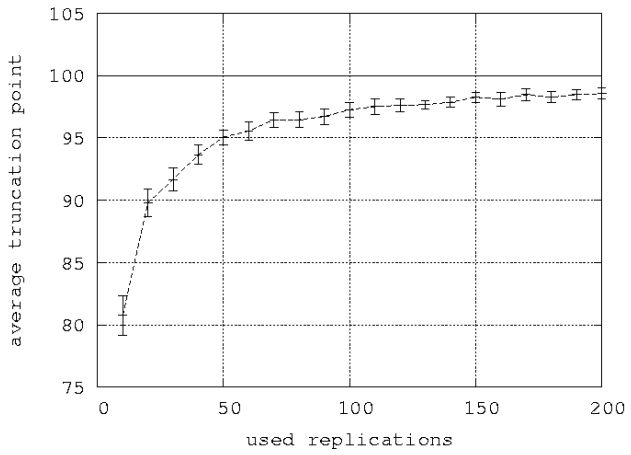


Figure 3: The quality of the estimate depends on the number of used parallel replications. This plot shows the analysis for process A with different numbers of replications.

CONCLUSIONS

We introduced an efficient algorithm for the estimation of the truncation point. The worst case time complexity of this algorithm is limited by $O(np\log(p))$ and is, therefore, substantially faster than the previous version of the algorithm.

This improvement is achieved without reducing the reliability of the estimates. This is experimentally shown on the basis of artificial output processes with a variety of different transient behaviours.

In the introduction it is pointed out that a truncation point estimation based on equation (1) can be used for many different performance measures, such as mean values, variances, quantiles or even rare-events. Equation (2) is in general only useful for mean value estimations.

However, limits concerning equation (1) are also discussed. It can be shown, that this equation does not always leads to the theoretically best values. This is simply caused by the approximation used. Using the equality instead of an approximation is no alternative, because this would lead in some cases (e.g. process C) to an infinite truncation point. Equation (1) can be used to reduce the initialization bias dramatically, but it cannot be used to eliminate it completely.

REFERENCES

[1] ALEXOPOULOS, C., AND GOLDSMAN, D. To batch or not to batch. *ACM Transactions on Modeling and Computer Simulation 14*, 1 (January 2004), 76–114.

[2] ARNS, M., EICKHOFF, M., FISCHER, M., TEPPER, C., AND VÖLKER, M. New Features in the ProC/B Toolset. In F. Bause (ed): Tools of the 2003 Illinois International Conference on Measurement, Modelling and Evaluation of Computer-Communication Systems. Technical Report 781, Uni Dortmund, Fachbereich Informatik, 2003.

[3] BAUSE, F., AND EICKHOFF, M. Initial transient period detection using parallel replications. *Proc. of the 14th European Simulation Symposium* (2002), 85–92.

[4] BAUSE, F., AND EICKHOFF, M. Truncation point estimation using multiple replications in parallel. *Proc. of the Winter Simulation Conference* (2003), 414–421.

[5] CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. *Introduction to Algorithms*. MIT Press, 1994.

[6] EICKHOFF, M. Statistische Auswertung und Erkennung der stationären Phase in der Simulation zustandsdiskreter Systeme. Diploma Thesis, Uni Dortmund, Fachbereich Informatik, 2002.

[7] FISHMAN, G. S. *Discrete-Event Simulation*. Springer, 2001.

[8] GAFARIAN, A. V., ANCKER, C. J., AND MORISAKU, T. Evaluation of commonly used rules for detecting steady state in computer simulation. *Naval Research Logistics Quarterly* (1978), 511–529.

[9] GHORBANI, B. The issue of initial transient in sequential steady-state simulation. Master thesis, University of Canterbury, Department of Computer Science, 2004.

[10] HAMILTON, J. D. *Time Series Analysis*. Princeton University Press, 1994.

[11] LAW, A. M., AND KELTON, W. D. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 2000.

[12] LEE, J.-S. R., MCNICKLE, D., AND PAWLIKOWSKI, K. Initial transient period detection for steady-state quantile estimation. *Proceedings of the Summer Computer Simulation Conference* (2000), 169–174.

[13] PAWLIKOWSKI, K. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys 22* (June 1990), 123–170.

[14] PAWLIKOWSKI, K., YAU, V., AND MCNICKLE, D. Distributed stochastic discrete-event simulation in parallel time streams. *Proc. of the Winter Simulation Conference* (1994), 723–730.

[15] WELCH, P. D. The statistical analysis of simulation results. *In The Computer Performance Modeling Handbook, ed. S. Lavenberg, Academic Press* (1983), 268–328.

[16] WHITT, W. The efficiency of one long run versus independent replications in steady-state simulation. *Management Science 77*, 6 (June 1991), 645–666.