"AN EXPERIMENTAL ANALYSIS OF STEADY STATE CONVERGENCE IN SIMPLE QUEUEING SYSTEMS: IMPLICATIONS FOR FMS MODELS"

by

R.P.A. NUYENS*
N.M. VAN DIJK**
L.N. VAN WASSENHOVE***
and
E. YÜCESAN****

93/51/TM

* Research Assistant, at INSEAD, Boulevard de Constance, Fontainebleau 77305 Cedex, France.

** Professor at Universiteit Van Amsterdam, The Netherlands.

*** Professor of Production and Operations Management at INSEAD, Boulevard de Constance, Fontainebleau 77305 Cedex, France.

**** Associate Professor of Production and Operations Management, at INSEAD, Boulevard de Constance, Fontainebleau 77305 Cedex, France.

Printed at INSEAD, Fontainebleau, France
An experimental analysis of steady state convergence in simple queueing systems:

Implications for FMS models

July 7th, 1993

R.P.A. Nuyens\textsuperscript{1},
N.M. Van Dijk\textsuperscript{2}, L.N. Van Wassenhove\textsuperscript{1}, E. Yücesan\textsuperscript{1}

Abstract

Queueing models are widely used for performance analysis of Flexible Manufacturing Systems (FMS). Most studies are done under steady state assumptions. The inherent flexibility of these systems in terms of input, volume or mix makes such an assumption hard to justify. In this paper, we investigate the convergence rate to steady state in simple queueing models that are often used to depict FMS's. The study reveals that steady-state conditions are not achieved during the execution of batches of moderate size. We therefore conclude that equilibrium models are appropriate for studying long-term planning problems, but, for the operational control of FMS's, it is highly desirable to study the transient behaviour of the system. On a broader scope, our investigation also shows that steady-state assumptions used in the analysis of any inherently transient system should be carefully validated.

INSEAD, Fontainebleau, France
Universiteit van Amsterdam, Amsterdam, The Netherlands
I Introduction

Continuous-time Markov chains provide a powerful framework for modelling and evaluation purposes in telecommunication systems, computer networks, and manufacturing. Increasingly, congestion models are used in depicting Flexible Manufacturing Systems (FMS). Transient analysis has therefore become more important, particularly for the analysis of operational control problems.

Explicit expressions are typically not available for transient characteristics. As exact computation rapidly becomes expensive, approximate procedures are used. Error bounds on the accuracy of such approximations, however, are rarely reported (see Van Dijk, 1991 for such bounds).

A more common approach is to use the readily available steady-state results. Such an approximation can be useful in that invocation of the steady-state assumption may permit an otherwise unobtainable, intuitively understandable, analytical characterization of the system behaviour. However, any attempts to extrapolate steady-state results to real systems, in which stationarity is not guaranteed a priori, should be carefully validated.

A Flexible Manufacturing System (FMS) is an automated manufacturing system consisting of a set of work stations linked by a material handling system and operated under computer control. The general philosophy behind an FMS is to combine the efficiency of transfer lines and the versatility of job shops so that the best features of both environments can be utilized in manufacturing products of medium volume and of modest variety. FMS's possess important advantages such as reduced lead times, consistent quality and increased flexibility (e.g. input, volume, routing). However, the fixed cost of acquiring such systems is often high; thus it is crucial to design and operate these systems at or near their 'optimal' point.

Analytical models play an important role in the study of these systems [Buzacott and Yao, 1986; Buzacott and Shantikumar, 1980; Cavaille and Dubois, 1982; Ho and Cao, 1983; Solberg, 1977; Stecke and Solberg, 1985; Suri and Leung, 1987; Yao and Buzacott, 1985a,b]. Although it is not likely that a single model can capture the complex nature of an FMS, analytical models may be used to get rough approximations in early design stages. For detailed design and operation stages, however, detailed mathematical modelling and optimization is typically impossible due to the inherent complexity and stochastic nature of these systems.

Queueing models represent the most popular approach in studying FMS's. An inherent assumption in queueing analysis is that parameters and other features of the system do not vary over time and that after a certain time the system reaches statistical equilibrium, the steady state. This is a reasonable assumption if it can be argued that any changes over the relatively limited period of interest are negligible. However, in everyday life, experience indicates that many phenomena are grossly nonstationary. Nevertheless, models based on a steady-state approximation can be useful in obtaining a characterization of the system behaviour, which may otherwise be difficult to capture.

In the study of FMS's there has been much emphasis on steady-state system behaviour; although one could argue that the inherent flexibility of these systems in terms of input, volume or routing makes the justification of the steady-state assumption hard to justify a priori. We feel that the error induced by using stationary models to analyze
potentially nonstationary systems, like some FMS's, seriously impairs the validity of these models in evaluating such systems. Equilibrium models are appropriate for investigating long-term planning problems, but for the operational control of FMS's it may be highly desirable to also study the transient behaviour of the system.

In this paper we investigate experimentally the convergence rate to steady state in simple queueing models that are often used to model FMS's. The study is anchored on an investigation via simulation of the transient behaviour of several basic queueing models through computer simulation. The way we measure the convergence rate is by determining the end of the initial transient period of the simulation output, i.e. the truncation point. We also try to indicate how various parameters influence the convergence rate. The method of truncation is straightforward and it combines visual inspection of moving averages with a confidence interval for the steady-state measure. Most of our results confirm intuition; however, these results provide quantitative insights into the impact of various system parameters and their interactions on system performance. For instance, the convergence rate decreases with increasing capacity of the queues and with increasing system complexity i.e. the number of stations, the number of different routings etc. The steady-state measure also has an influence on the convergence rate. The use of sojourn time instead of throughput as the steady-state measure significantly decreases the convergence rate. The capacitated systems warm up quite quickly: mostly within the first 100 service completions. But as soon as capacity increases and more workstations are included, the truncation points grow considerably. Thus, in a complex FMS environment it is very unlikely that the system warms up before the 100th service completion. So, the use of batches of 100 or even fewer entities in FMS's does not allow the system to warm up within the length of the batch. Therefore the evaluation of FMS's through steady-state models should be handled with care.

This paper is organized as follows: the models we have used are described in section II, together with the experimental design for the study and the simulations. Here, the determination of the truncation point is also described. In section III, the results are presented. Section IV discusses the results and section V offers some conclusions.
II Methodology

2.1 The Study

Simple queueing models are simulated for different configurations; for each of these configurations a truncation point is determined. The truncation point in a simulation output is the index of the last observation before steady state is reached. The impact of the parameters in the queueing models on the truncation point (i.e. convergence rate to steady state) is assessed through a $2^k$-factorial design. For completeness, this method is described in Section 2.1.1. The methods of determining the truncation point, which are used in this study, are described in Section 2.1.2.

Queueing systems with three different configurations have been studied, as shown in Figure 1.

- Single server queue
- Two queues in tandem (Tandem)
- Four queues in a split and merge configuration (Network)

*Figure 1. Basic queueing configurations.*

Two different sets of finite-capacity queueing models were implemented. The difference between these two sets of models is the arrival process. In the first set of models the first queue is uncapacitated and will be filled with all the entities of the first batch at time zero. This may represent a batch production environment. In the second set of models only finite capacity queues are used. Here the arrival process follows a Poisson process. In both sets of models the service times in the queues are independent and exponentially distributed. Between two consecutive queues there is (service) blocking: if an entity completes its service while the destination queue is full (all waiting places are occupied), then it has to wait in front of the server, until the next queue finishes its current service. So until that point the former server is blocked and there is no other service in that queue. In addition to this type of blocking, the second model can deny access to entities upon arrival to the first queue, if that queue is full.

The three queueing systems will be run under different parameter configurations. In each queueing system three parameters are varied: capacity of a queue, number of servers in a queue and utilization or service rate of a server. The last parameter depends
on which model is used. The capacity of a queue is the number of waiting places including the places available for the entities in service. All the queues in the system are the same, except for the first queue for the first set of models, because it has infinite capacity. The performance measure for the first set of models is throughput and for the second set of models throughput and sojourn time.

The models process identical batches. The processing of the batches for the first set of models begins with the placement of the entire batch in the first queue. In reality this would mean that a batch will be processed as soon as the entire batch is ready to be processed by the system (production line). So the first queue never has to wait for new products during the processing of a batch. The other queues are all empty at time zero. In the second set of models, on the other hand, the process starts with a totally empty system and the entities of a batch are processed as soon as an entity arrives at the production line. The arrival process follows a Poisson process with parameter $\lambda$ and the first entity of a batch arrives after an exponentially distributed time period with mean $1/\lambda$. The system at the beginning of a new batch is empty, because it is reasonable to assume that a new batch is started only if the last batch has been processed totally and thus has left the system.

2.2 Experimental Design

There are two settings for each parameter: a low and a high value. Since the two sets of models differ in arrival process, the parameter values for each model are specified separately below (Table 1).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low value</th>
<th>High value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service rate of server</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Capacity</td>
<td>1*</td>
<td>4</td>
</tr>
<tr>
<td>Number of servers</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1a. Experimental design for first model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low value</th>
<th>High value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>20%</td>
<td>90%</td>
</tr>
<tr>
<td>Capacity</td>
<td>1*</td>
<td>4</td>
</tr>
<tr>
<td>Number of servers</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1b. Experimental design for second model

(*) 2 in cases of 2 servers

In spite of an increase in the number of servers the value of the utilization is kept constant by adjusting the service rate.

In a $2^k$-factorial design [Law & Kelton, 1991, p.659-670] the effect of each parameter of a model on the response of the simulations of that model is determined. The response of the simulations could be one of the steady-state measures of the model. The effect of a parameter on the response is the average increase or decrease in the response, when changing only that parameter from its low to its high value, while keeping all other parameters constant. This effect is called the main effect of a parameter.
In this study, the response of interest is the truncation point i.e. the index of the last service completion before steady state. In the case of three parameters, each system is simulated for 8 \(2^3\) different configurations. The following matrix (Table 2) is the representation of all 8 configurations with the corresponding responses. A minus sign is associated with the low level and a plus sign with the high level.

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Utilization</th>
<th>Number of servers</th>
<th>Response (truncation point)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>R1</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>-</td>
<td>R2</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>-</td>
<td>R3</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>-</td>
<td>R4</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>+</td>
<td>R5</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
<td>R6</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>R7</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>R8</td>
</tr>
</tbody>
</table>

Table 2. Representation of all configurations

Thus, the average change in the response due to a change in capacity is:

\[
\frac{(R_2 - R_1) + (R_4 - R_3) + (R_6 - R_5) + (R_8 - R_7)}{4}
\]

With a 2\(^k\)-factorial design it is also possible to measure the impact of the interaction of two parameters. This effect is called the two-way interaction effect. The two-way interaction effect of parameters 1 and 2 is defined as: half of the difference between the average effect of parameter 1 when parameter 2 is at its '+' level (all the other parameters are held constant) and the average effect of parameter 1 when parameter 2 is at its '-' level. For example, in Table 2 the two-way interaction effect of capacity and utilization is:

\[
0.5 \left\{ \frac{(R_2 - R_1) + (R_8 - R_7)}{2} - \frac{(R_2 - R_1) + (R_8 - R_7)}{2} \right\}
\]

2.3 Simulation

The simulations are run with the simulation package SLAMSYSTEM version 2.1 [Pritsker, 1989]. A simulation run is made up of 10 cycles of 200 service completions. A service completion is the completion of service in the last queue, so the entity leaves the system. A batch size of only 200 entities is relatively small compared to the batch sizes used in steady-state simulations, but the objective of this study is to investigate the steady-state assumption in models of FMS's. One of the specific characteristics of an FMS is batch sizes, which tend to be far smaller than 200 parts. A new cycle starts, as soon as the last cycle has been completely processed. This means, for the first set of models, that a
new batch of 200 entities is placed in the first queue as soon as the last entity of the last batch has left the system. For the second model the arrival process is suspended, when 200 entities have entered the system. The execution continues until the last entity has left the system.

One simulation run will thus result in 10 virtually independent replications of one batch. Each simulation run is replicated 5 times with different seeds, yielding 50 virtually independently-seeded replications, each for 200 service completions. Each replication consists thus of 200 values for the throughput and of 200 values for the sojourn time (sojourn time only for the second set of models). The output series of these 50 replications are averaged across replications to reduce variance. So the final output series consists of 200 values for each performance measure.

The throughput of an entity is the reciprocal value of the difference between its service completion and the service completion of its predecessor. For the first entity of the first batch the throughput is defined as the reciprocal value of the difference between its service completion time and the start time of the simulation (time zero), which includes the exponential time before the first entity enters the system. The throughput for the first entities of the following batches is the reciprocal value of the time between their service completions and the service completion of the last entity of the preceding batch. The sojourn time of an entity is defined as the time between entering and leaving the system.

2.4 Criteria for Truncation

A general method for determining a truncation point in the output series of computer simulations is not available. There are a lot of different heuristic methods, but all of them have some disadvantages [Gafarian et al, 1978]. Two different types of tests are performed in this paper. The first is the simplest and most general initialization bias test; visual inspection of the simulation output. This graphical method is due to Welch (1981, 1983) and its goal is to determine when the transient mean curve flattens out at the steady-state level. Welch's procedure is based on making n independent replications of the simulation of length m and then averaging across these replications. To smooth out the high-frequency oscillations in the resulting output series, \( Y_1, Y_2, \ldots, Y_m \), moving averages are used. The moving average \( Y_i(w) \) (where \( w \), the windowsize, is a positive integer such that \( w \leq m/2 \)) is defined as follows:

\[
Y_i(w) = \begin{cases} 
\frac{\sum_{s=-w}^{w} Y_{i+s}}{2w + 1} & i = w + 1, \ldots, m - w \\
\frac{\sum_{i-1}^{m-w} Y_{i+s}}{2i - 1} & i = 1, \ldots, w
\end{cases}
\]
The series $Y_i(w)$ are plotted for $i=1,2,...,m-w$. Visual inspection of the moving averages allows one to determine when $Y_1(w), Y_2(w),...$ appears to have 'settled down'. A window size of 50 is used in this study. To help determining convergence, a 90%-confidence interval for the mean of $Y_i$ has been added to the moving average plot. This confidence interval is based on the last 150 observations of the raw data (that is, before taking moving averages). The criterion for determining a truncation point is that the moving averages $Y_i(50)$ should stay within the confidence interval after the truncation point in order to conclude that the system is in steady state after that point.

The other initialization bias test [Schruben et al, 1983] detects initialization bias in the mean of a simulation output series using a hypothesis testing framework. The null hypothesis is that the output mean does not change throughout the simulation run (i.e. no initialization bias in the simulation output). This test was initially chosen, because it is very easy to implement and appears to be both robust and powerful. The output of the test is a t statistic. The test consists of the following steps:

1. Estimation of variance and degrees of freedom.

   Use the last $r$ observations from the output series $Y_1, Y_2,..., Y_m$ for the estimation of $\sigma^2$. Relabel these $r$ observations into $Y_1, Y_2,..., Y_r$. Divide the series $Y_1, Y_2,..., Y_r$ in $k$ batches of equal size. Compute [Fishman,1973, p.279-289]:

   \[
   \sigma^2 = \frac{r}{k(k-1)} \sum_{j=1}^{k} (\bar{Y}_{(j)} - \bar{Y}_r)^2
   \]

   \[
   d = k
   \]

   where

   $\bar{Y}_{(j)} =$ sample mean of batch $j$

   $\bar{Y}_r =$ sample mean of output series $Y_1, Y_2,..., Y_r$

   $d =$ degrees of freedom of t statistic

2. Calculation of the test statistic $T$.

   \[
   T = \frac{\sqrt{45} \sum_{j=1}^{m} (1 - \frac{j}{m}) \sqrt{\frac{(\bar{Y}_m - \bar{Y}_j)}}{m^{\frac{3}{2}} \sigma}}{\sigma}
   \]

   where
The null hypothesis will be rejected if the value of $T$ is larger than the critical value $t_{1-\alpha/2, d}$.

The significance level $\alpha$ for this two-sided test is 0.10. The output series was divided into 5 batches. The critical value for $\alpha=0.10$ and $d=5$ is 2.015. The test can also be performed on the simulation output series after truncation, i.e. only on the remaining data. Thus, in that case the second part of the test needs some trivial adjustments.

This test is used to decide whether there is any initialization bias in the remaining data: first the test is performed on the entire simulation output and then the first observation is truncated and after that the test is performed on the remaining data. So in each subsequent step the first observation of the remaining data is truncated and the test is performed on the remaining data. The variance estimator stays the same throughout the entire testing procedure. This procedure continues until 100 observations are left. The shape of the resulting plot of the test values was expected to be high in the beginning suggesting the presence of initial bias and to subsequently decline and stay below the critical value of 2.015. A well-behaved example is observed from the test results for a queueing network with split-and-merge configuration, where the first queue is uncapacitated, the other queues have capacity 4, there is one server in each queue and the service rate of the servers is 10. Figure 2a shows the moving averages of this simulation and Figure 2b shows the corresponding test results. The test results start above 2.015 and they decline quickly and stay below the critical value for the rest of the simulation output. So only in the very beginning of the simulation output the test detects some initialization bias. An example where the test behaves not so well is given in Figures 3a and 3b. This case deals with the same system as given above, only the service rate is now 20 instead of 10 and the capacity is 1 in each queue except for the first one. In this case the test values stay above the critical value almost throughout the whole run. This could indicate that the system hasn't warmed up yet, but Figure 3a, the moving averages of the same simulation, shows that the system has already reached steady state. Another concern about the test is that the test assumes that the system has warmed up much earlier than where the moving averages flatten out (Figure 2a and 2b). These two major drawbacks of the initialization bias test in our applications made us decide to use exclusively the visual inspection of the moving averages to conclude that steady state has been reached as soon as all moving averages after truncation stay within the 90%-confidence interval.
III Results

This section deals with the presentation of the results. First the results concerning the convergence are presented: for both sets of models, the truncation point in each configuration is given. The second part of this section gives an overview of the different parameter effects, which are determined by a factorial experiment. The implications of the results are discussed in Section IV.

3.1 Convergence Results

Table 3 contains the truncation points for each configuration for the first set of models. The truncation points are based on the simulation runs of 200 entities. For the
configurations with only one server these simulations were long enough to determine a truncation point, but for the two-server configurations more service completions were needed to reach steady state. So the truncation points for the two-server configurations are based on simulation runs of 800 entities. After several hundred service completions, most moving averages for the two-server configurations stay within the confidence interval, but they do not look stable. For the single-server systems no results have been obtained, because, due to the uncapacitated first queue, the single-server systems cannot be included in the $2^k$-factorial design. It is, however, a well-known fact that M/M/1/∞ simulations have a long warm-up period. According to the plots some of the configurations have not reached steady state during the run and therefore they are marked with an '-' in the tables.

<table>
<thead>
<tr>
<th>One Server</th>
<th>Tandem</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Service Rate</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 3. Truncation points for first set of models

In Table 4a the truncation points for the second set of models are given with respect to throughput. Table 4b contains the truncation points based on sojourn time for the second model. All the simulation runs of the second model are for 200 service completions.

<table>
<thead>
<tr>
<th>One Server</th>
<th>M/M/1</th>
<th>Tandem</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Utilization</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>20%</td>
<td>50</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>90%</td>
<td>19</td>
<td>20</td>
<td>39</td>
</tr>
<tr>
<td>20%</td>
<td>13</td>
<td>39</td>
<td>72</td>
</tr>
<tr>
<td>90%</td>
<td>21</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>20%</td>
<td>13</td>
<td>39</td>
<td>72</td>
</tr>
<tr>
<td>90%</td>
<td>21</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>20%</td>
<td>13</td>
<td>39</td>
<td>72</td>
</tr>
<tr>
<td>90%</td>
<td>21</td>
<td>21</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 4a. Throughput truncation points for second set of models
<table>
<thead>
<tr>
<th>One Server</th>
<th>M/M/1</th>
<th>Tandem</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Utilization</td>
<td>20%</td>
<td>34</td>
<td>19</td>
</tr>
<tr>
<td>90%</td>
<td>20</td>
<td>50</td>
<td>110</td>
</tr>
<tr>
<td>Two Servers</td>
<td>M/M/2</td>
<td>Tandem</td>
<td>Network</td>
</tr>
<tr>
<td>Capacity</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Utilization</td>
<td>20%</td>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td>90%</td>
<td>16</td>
<td>54</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4b. Sojourn time truncation points for second set of models

3.2 Sensitivity

In this section the main and two-way interaction effects are presented. The parameters and their corresponding low and high values have been discussed in Sections 2.2 and 2.3. Because we have three different systems in both sets of models we introduce a qualitative parameter: the system complexity. It indicates which of the three queueing systems is used in the simulation run. A high value is always associated with the more complex system.

Many simulation runs for the first set of models with two servers in each queue do not seem to settle down, even after several hundred service completions. Therefore the calculations of the effects for the first model are only based on the one-server configurations. Table 5 contains the main and the two-way interaction effects for the first model. The values on the diagonal are the main effects, the two-way interaction effects are symmetric and thus displayed only once.

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Service Rate</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>+40.25</td>
<td>+5.75</td>
</tr>
<tr>
<td>Service Rate</td>
<td>+13.25</td>
<td>+16.75</td>
</tr>
<tr>
<td>System</td>
<td>+29.25</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Parameter-effects in first set of models

For the second set of models there are two more parameters: number of servers and a qualitative parameter that indicates which steady-state measure is used. The high value of this parameter has been arbitrarily associated with throughput, the low value with sojourn time. Table 4b shows that some of the simulation runs have not warmed up within the first 200 service completions. For these runs we assume that the truncation point is 200. Since there are three different systems (i.e. single server, tandem and network), the $2^k$-factorial design has to be performed for each pair of these systems. In Tables 6a-6c all the main and two-way interaction effects for the second set of models are given.
<table>
<thead>
<tr>
<th>Single Server-Tandem</th>
<th>Capacity</th>
<th>Utilization</th>
<th># of Servers</th>
<th>System</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>+19.3125</td>
<td>+7.6875</td>
<td>+3.9375</td>
<td>+0.3125</td>
<td>-16.5625</td>
</tr>
<tr>
<td>Utilization</td>
<td>+39.3125</td>
<td>+5.4375</td>
<td>+47.5625</td>
<td>-33.5625</td>
<td></td>
</tr>
<tr>
<td># of Servers</td>
<td>+13.3125</td>
<td>+11.8125</td>
<td></td>
<td></td>
<td>-2.0625</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>+42.9375</td>
<td></td>
<td></td>
<td>-24.4375</td>
</tr>
<tr>
<td>Measure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-33.1875</td>
</tr>
</tbody>
</table>

Table 6a. Single server - Tandem parameter-effects in second set of models

<table>
<thead>
<tr>
<th>Single Server-Network</th>
<th>Capacity</th>
<th>Utilization</th>
<th># of Servers</th>
<th>System</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>+16.625</td>
<td>+6.750</td>
<td>+0.875</td>
<td>-2.375</td>
<td>-16.875</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+30.000</td>
</tr>
<tr>
<td>Measure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-17.000</td>
</tr>
</tbody>
</table>

Table 6b. Single server - Network parameter-effects in second set of models

<table>
<thead>
<tr>
<th>Tandem-Network</th>
<th>Capacity</th>
<th>Utilization</th>
<th># of Servers</th>
<th>System</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>+54.1875</td>
<td>-2.6875</td>
<td>-32.6875</td>
<td>-43.9375</td>
<td></td>
</tr>
<tr>
<td># of Servers</td>
<td>-6.4375</td>
<td>-31.5625</td>
<td>+25.4375</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>-12.9375</td>
<td></td>
<td>+16.1875</td>
<td></td>
</tr>
<tr>
<td>Measure</td>
<td></td>
<td></td>
<td></td>
<td>-41.4375</td>
<td></td>
</tr>
</tbody>
</table>

Table 6c. Tandem - Network parameter-effects in second set of models

IV Discussion

In this section the results of the two models are discussed separately. The discussion is held for each single parameter. In section V this discussion will be summarized and final conclusions will be drawn.

In contrast to steady-state analysis, explicit expressions are scarce for transient characteristics. Cinlar [1975, p.378] discusses convergence rates for aperiodic irreducible Markov matrices. It has also been suggested for Markov chains with a bounded regeneration time, M, that the chain is in steady state at time M. Consequently, the initial transient effects can be eliminated by deleting the first M time units of the simulation.

Glynn and Iglehart [1988], however, show that boundedness of regeneration times does not imply that the chain is in steady state in finite time. It is therefore useful to investigate experimentally the transient behaviour of the system and to observe the impact of various parameters on the rate of convergence.

It is anticipated that the size of the state space and the concentration of the probability mass over that state space (or equivalently over the associated transition matrix) are the two principal factors affecting the convergence rate. Roughly speaking, the
larger the state space of a system, the longer the initial transient period will be. This is because, for an aperiodic irreducible chain, a larger state space may result in a longer regeneration time.

The concentration of the probability mass may have varying degrees of influence. For instance, if it is concentrated in one small part of the state space, the system will spend more time within this portion of the state space; the rest of the state space will therefore be visited quite infrequently. Thus a concentrated probability mass could be seen as a reduction factor for the initial transient.

In Figure 4 the state space of a capacitated tandem queue is shown. The X- and Y-axes contain the number of entities in each queue. For a high utilization configuration the transition probabilities are primarily concentrated on the upper righthand corner and the other states are visited with low or even negligible probabilities. The probability mass in a low utilization configuration is concentrated in the lower lefthand corner. The convergence rate appears slowest when the middle area also contains a considerable probability mass.

![Figure 4. State space of a tandem queue.](image)

On the other hand, changes in arrival and/or service rates may affect not only the distribution of the probability mass over the state space by altering the transition probabilities, but may also have an impact on the sojourn times. That is, the time spent by the system in a given state may also change. Therefore, modifications in arrival and/or service rates may have an unpredictable effect on the convergence rate.

Most of our experimental findings confirm the above intuition. In a few instances, however, the results seem to be counterintuitive at first glance. They are nevertheless explained by considering potential interactions among various parameters.

## 4.1 Model 1

For the purpose of illustration in this first model, we have restricted our investigations to single-server stations. Clearly, as will be illustrated in the second model, similar results can be obtained with more servers in which case the convergence rate will be increased.

### 4.1.1 Capacity

The truncation points for the one-server configurations in Table 3 show that a bigger capacity in each queue increases the length of the initial transient period. This is also shown by the main effect of capacity in Table 5. The magnitude of the main effects
gives some insights into the quantitative impact of each of the parameters. From Table 5 it is observed that capacity has the biggest impact on the truncation point (+40.25). As an increase in the capacity has a direct influence on the size of the state space, a bigger capacity creates more states and thus it will take more time to warm up the system.

4.1.2 System Complexity

Another observation from Table 3 is that a more complex system (i.e. network configuration instead of tandem configuration) also prolongs the transient period. More complex systems mean more queues and therefore more possible states the system can visit. In addition, the number of possible transitions is larger in more complex systems. This impact of the system complexity is confirmed by the main effect of system complexity in Table 5 (+29.25).

4.1.3 Number of Servers

One can conclude from the truncation points in Table 3 that an increase in the number of servers in each queue causes the warm up period to be remarkably longer. Run lengths of 800 service completions were not sufficient for most configurations to reach steady state. The moving averages, shown in Figures 5 and 6, keep oscillating heavily after several hundred service completions. Note that an increase in the number of servers also increases the number of states. A larger state space, in turn, results in a longer transient period.

4.1.4 Service Rate

The influence of the service rate on the truncation point varies among different system configurations. In most cases, lower service rates lead to faster convergence. This
is because, in a capacitated queueing system, starting from an 'empty and idle' state, typical congested states are achieved more quickly with lower service rates.

The two-way effects between service rate and capacity (+5.75) and between service rate and system complexity (+16.75) should also be taken into consideration. These results imply that an increasing service rate decreases the convergence rate more in systems with higher capacity and with more complex configurations than in systems with lower capacity and simpler configurations. This is because, if there are only four possible states and the service rate is changing, each state will still have a relatively high probability, whereas in a situation with sixteen possible states and a changing service rate, there are some states which could be neglected and thus, the convergence rate would change significantly.

4.2 Model 2

For the second set of models it was much easier to determine truncation points based on the moving averages for the throughput than it was for the first model, since the moving averages showed a clear distinction between the transient and the steady-state periods. Especially after truncation these systems were oscillating less than the systems in the first set of models. The two-server configurations, which gave some problems in the first model, were as easy to truncate as the one-server configurations.

For the second set of models we focus on three different kinds of effects, namely the effects that occur if the single server is replaced by the tandem configuration, if the single server is replaced by the network configuration and if the tandem is replaced by the network configuration. In order to draw general conclusions about the influence of parameters on the truncation point, the corresponding main effects in these three cases should be at least of the same sign.

4.2.1 Capacity

Tables 4a and 4b show that an increase in the capacity seems to decrease the convergence rate, but there are a few configurations where the opposite is true. The three main effects for capacity in Tables 6a, 6b and 6c are all positive, thus an increase in the capacity of the queues slows down the convergence of the system. This corresponds to the assertion that an increase in the number of states slows down the convergence rate. The cases where the truncation point decreases with capacity could be explained partly by the interaction between capacity and other parameters such as utilization, number of servers and system complexity.

4.2.2 System Complexity

The influence of system complexity confirms our assertion that larger state spaces (due to more complex models) yield longer transient periods. This relationship is strongly present when the single-server system is compared with the tandem queue and the network (+42.9 and +30.0 respectively). This relationship, however, is not captured in the comparison of the tandem system with the network. This could be due to the smoothing
effect symmetric branching probabilities in the network may have on the measures of performance.

4.2.3 Number of Servers

An increase in the number of servers slows down the convergence process based on the moving averages for the throughput (Table 4a). In all configurations, except one, the truncation point increases with an increase of the number of servers. But if the truncation points are based on the moving averages for the sojourn time, then there are far more configurations where the truncation point decreases with the number of servers. Tables 6a, 6b and 6c show that the main effects of the number of servers fluctuate throughout the three different cases.

There are two different consequences of an increase in the number of servers. First, the number of states increases; thus this is a reason to believe that the convergence process slows down with the number of servers. Secondly, with two servers working in each workstation, the jobs can be processed more smoothly, although the utilization of the workstation does not change. This can be observed quite well in Table 4b (sojourn time), where there are some configurations with two servers which warm up much quicker than the ones with one server. Thus with two servers the sojourn time has much less variation than with one server. The two-way interaction effects between the number of servers and the steady-state measure show the same impact. These effects are -2.1, +18.3 and -25.4. The positive effects dominate the small negative one. A positive two-way interaction effect between the number of servers and the steady-state measure implies, that increasing the number of servers decreases the convergence rate more when throughput is used as a measure of performance, than when sojourn time is used. The increasing number of servers speeds up the convergence process when sojourn time is used.

4.2.4 Utilization

The influence of utilization is strongly dependent on the specific configuration and thus on the interaction between all the parameters. Figure 7 shows the dependency of the truncation point on the utilization. For a utilization range between 20% and 110% the truncation points have been determined for the tandem system with capacity 4 and one server in each queue. Figure 7 shows that in the low utilization configurations the truncation point increases with the utilization, but after a certain point (60-70%) the truncation point starts decreasing slowly. This is because the probability mass is concentrated in a smaller portion of the state space in both high and low utilization ranges, as depicted in Figure 4. The peak in Figure 7 corresponds to the case where the probability mass is distributed more evenly throughout the state space.
An important two-way effect is the interaction between capacity and utilization. In all three cases this effect is positive, which means that the influence of utilization on truncation point is more significant in systems with bigger capacities.

4.2.5 Steady-State Measure of Performance

The use of the moving averages of the sojourn time makes it more difficult to determine the end of the transient period. Consider, for instance, Figures 8 and 9. Figure 8 shows the moving average of the throughput for the tandem queue with two servers in each queue, capacity two and 90% utilization. Figure 9 shows the moving average of the sojourn time for the same configuration. According to the moving averages of the throughput the system would have been warmed up after the 50th service completion. However, no truncation point could be determined for the moving averages of the sojourn time.
The parameter for the steady-state measure has a negative effect. Since throughput has been associated with the high level, a negative effect means that the use of the throughput as the steady-state measure increases the convergence rate. It was also more difficult to determine the truncation points from the sojourn time moving averages. A possible explanation might be that sojourn time is the sum of several stochastic processes, i.e. sum of the waiting and processing times in each queue and that it therefore has a bigger variability than throughput.

V Conclusions

The objective of this study was to investigate the convergence rate to steady state in simple queueing models that are often used in modeling FMS. To this end, simulation experiments have been conducted for different configurations of simple queueing systems. The convergence rate is measured by determining the end of the initial transient period of the simulation output. The method of truncation is easy to implement and performed in a satisfactory way throughout the experiment. This method combines a confidence interval around the estimator of the steady-state measure with moving averages of the same steady-state measure. This approach turned out to be useful, because it combines visual inspection of the moving averages with a quantitative measure for the variation of the simulation output. The results of our simulations assert that:

- convergence rate decreases with capacity,
- convergence rate decreases with system complexity, and
- the choice of the performance measure influences the convergence rate.

For utilization and number of servers such clear-cut conclusions cannot be drawn, because these parameters can have different effects in different configurations. To analyze these dependencies further replications are needed in order to decrease the noise in the simulation output, that in turn, will increase the accuracy of the truncation points.

The length of the initial transient period in most of the cases is smaller than 100 service completions, but this length increases quickly with bigger capacities and with more workstations. Since only simple queueing systems have been investigated, one can extrapolate that in a real FMS, where complexity is much higher in terms of input, volume or routing, the transient period will probably be longer than 100 service completions and thus steady state might not be reached within the length of one batch. Steady-state analysis of FMS is therefore becoming less accurate as the batch sizes are being reduced to well below 100 pieces. Hence, in processing different batches during daily operations, an FMS may never achieve steady state. Consequently, the steady-state assumption of most FMS models may seriously impair the accuracy of the performance analysis. In the worst case, this may lead to a suboptimal operation which, in turn, would fail to yield the anticipated savings.

We should also point out that our analysis applies not only to the models of FMS, but to any situation where steady-state analysis is used to study an inherently transient system. Such an extrapolation should always be carefully validated.
Further research in this field will consist of the investigation of more complex queueing systems, especially combinations of the systems, which have been investigated in this paper. In these more complex systems different batches should also be processed in order to get a more realistic representation of a production environment. Detailed simulation of a real FMS will be difficult and time consuming, but it is definitely needed to validate our results in a real life environment.

Acknowledgement

The support of INSEAD through R&D project 2110R is gratefully acknowledged.

References


