# Deducing Queueing From Transactional Data: The Queue Inference Engine, Revisited

by

*Dimitris J. Bertsimas & Les D. Servi*

OR 212-90 April 1990

# Deducing Queueing from Transactional Data: the Queue Inference Engine, Revisited

Dimitris J. Bertsimas * L. D. Servi †

April 11, 1990

**Abstract**

Larson [1] proposed a method to statistically infer the expected transient queue length during a busy period in $O(n^5)$ solely from the $n$ starting and stopping times of each customer's service during the busy period and assuming the arrival distribution is Poisson. We develop a new $O(n^3)$ algorithm which uses this data to deduce transient queue lengths as well as the waiting times of each customer in the busy period. We also develop an $O(n)$ on line algorithm to dynamically update the current estimates for queue lengths after each departure. Moreover we generalize our algorithms for the case of time-varying Poisson process and also for the case of iid interarrival times with an arbitrary distribution. We report computational results that exhibit the speed and accuracy of our algorithms.

*Dimitris Bertsimas, Sloan School of Management and Operations Research Center, MIT, Rm. E53-359, Cambridge, Massachusetts, 02139.

†L. D. Servi, GTE Laboratories Incorporated, 40 Sylvan Road, Waltham, Massachusetts, 02254.
This work was completed while L. D. Servi was visiting the Electrical Engineering and Computer Science Department of MIT and the Division of Applied Sciences at Harvard University while on sabbatical leave from GTE Laboratories.

# 1 Introduction

In 1987, Larson proposed the fascinating problem of inferring the transient queue length behavior of a system solely from data about the starting and stopping time of each customer service. He noted that automatic teller machines (ATM's) at banks have this data, yet they are unable to directly observe the queue lengths. Also in a mobile radio system one can monitor the airwaves and again obtain times of call setups and call terminations although one is unable to directly measure the number of potential mobile radio users who are awaiting a channel. Other examples include checkout counters at supermarkets, traffic lights, and nodes in a telecommunication network. More generally, the problem arises when costs or feasibility prevents the direct observation of the queue although measurements of the departing epochs are possible. This paper discusses the deduction of the queue behavior only from this transactional data. In this analysis no assumptions are required regarding the service time distribution: it could be state-dependent, dependent on the time of day, with or without correlations. Furthermore, the number of servers can be arbitrary.

In Larson [1] two algorithms are proposed to solve this problem based on two astute observations: (i) the beginning and ending of each busy period can be identified by a service completion time which is not immediately followed by a new service commencement; (ii) a service commencement at time $t_i$ implies that the arrival time of the corresponding customer must have arrived between the beginning of the busy period and $t_i$. Furthermore, if the arrival process is known to be Poisson, then the a posteriori probability distribution of the arrival time of this customer must be uniformly distributed in this interval. In Larson [1] these observations are used to derive an $O(n^5)$ and an $O(2^n)$ algorithm to compute the transient queue lengths during a busy period of $n$ customers.

In this paper we propose an $O(n^3)$ algorithm, which estimates the transient queue length during a busy period as well as the delay of each customer served during the busy period. Moreover, we generalize the algorithm for the case of time-varying Poisson process to another $O(n^3)$ algorithm and also find an algorithm for the case of stationary interarrival times from an arbitrary distribution. We also develop an $O(n)$ on line algorithm to dynamically update the current estimates for queue lengths after each departure. This algorithm is similar to Kalman filtering in structure in that the current estimates for future queue lengths are updated dynamically in real time.

The paper is structured as follows: In Section 2 we describe the exact $O(n^3)$ algorithm and prove its correctness assuming the arrival process is Poisson. In Section 3 we generalize the algorithm for the case of time-varying Poisson process into a new $O(n^3)$ algorithm. In Section 4 we further generalize our methods to handle the case of an arbitrary stationary interarrival time distribution. In Section 5 we describe the algorithm to dynamically update queue lengths in real time. In Section 6 we report numerical results. We also examine the sensitivity of the estimates to the arrival process.

# 2 Stationary Poisson Arrivals

In this section we will assume that the arrival process to the system can be accurately modeled as a Poisson process with an arrival rate that it is unknown but constant within a busy period. The arrival rate, however, could vary from busy period to busy period. In sections 3 and 4 this assumption is relaxed. We consider the following problem: Given only the departure epochs (service completions) $t_i$, $i = 1, 2, \ldots, n$ during a busy period in which $n$ customers were served, estimate the queue length distribution at time $t$ and the waiting time of each customer. As we noted before we make no assumptions on the service time distribution or the number of servers.

## 2.1 Notation and Preliminaries

The following definitions are needed. Let

1. $n =$ the number of customers served during the last busy period (the end of the busy period is inferred from a service completion that is not immediately followed by a new service commencement).

2. $t_i =$ the time of the $i$th customer's service completion.

3. $X_i =$ the arrival time of the $i$th customer.(By definition $X_1 \leq X_2 \ldots \leq X_n$. )

4. $N(t) =$ the cumulative number of arrivals $t$ time units after the current busy period began.

5. $Q(t) =$ the number of customers in the system just after $t$ time units after the current busy period began.

6. $D(t) =$ the cumulative number of departures $t$ time units after the current busy period began.

7. $W_j =$ the waiting time of the $j$th customer.

8. $O(\vec{t}) =$ the event $\{X_1 \leq t_1, \ldots, X_n \leq t_n\}$.

9. $O'(\vec{t}, n) =$ the event $\{X_1 \leq t_1, \ldots, X_n \leq t_n \textit{ and } N(t_n) = n\}$.

Without loss of generality assume that the busy period starts at time $t_0 = 0$.

We observe the process $D(t)$ and wish to characterize the distribution of $N(t)$ given $D(t)$. From $N(t)$ properties of $Q(t)$ and $W_j$ can be directly computed. For example $Q(t) = N(t) - D(t)$. Suppose exactly $n-1$ consecutive service completions coincide with new service initiations at times $t_1, t_2, \ldots, t_{n-1}$, and a service completion takes place at time $t_n$ *without* a simultaneous new service initiations. Then, one knows that event $O'(\vec{t}, n)$ occurred, i.e., $X_1 \leq t_1, \ldots, X_n \leq t_n$ and $N(t_n) = n$. We want to compute the distribution of $N(t)$ conditioned on $O'(\vec{t}, n)$.

To do this we note (Ross [3]) that the conditional joint density for $0 \leq x_1 \leq x_2 \ldots \leq x_n$

$$f(X_1 = x_1, \ldots, X_n = x_n | N(x_n) = n) = \frac{n!}{x_n^n}. \tag{1}$$

As a result,

$$Pr\{X_1 \leq t_1, \ldots, X_n \leq t_n | N(t_n) = n\} = \frac{n!}{t_n^n} \int_{x_1=0}^{t_1} \int_{x_2=x_1}^{t_2} \cdots \int_{x_n=x_{n-1}}^{t_n} dx_1 \ldots dx_n. \tag{2}$$

Equation (2) is an example of an integral that encapsulates the a posteriori information of the arrivals times. This information combined with the known departures times lead to a posteriori estimates of the queue lengths and waiting times. More generally, as will be shown in Section 2.2 and proven in Section 2.3 to evaluate this type of integral efficiently we must evaluate the following related integrals

$$H_{j,k}(y) = \int_{x_1=0}^{t_1} \int_{x_2=x_1}^{t_2} \cdots \int_{x_j=x_{j-1}}^{t_j} \int_{x_{j+1}=x_j}^{y} \cdots \int_{x_k=x_{k-1}}^{y} dx_1 \ldots dx_k \tag{3}$$

and

$$F_k(y) = \int_{x_k=0}^{y} \int_{x_{k+1}=x_k}^{t_{k+1}} \cdots \int_{x_n=x_{n-1}}^{t_n} dx_k \ldots dx_n \tag{4}$$

at $y = t_j$.

Define

$$h_{j,k} = H_{j,k}(t_j) = \int_{x_1=0}^{t_1} \int_{x_2=x_1}^{t_2} \cdots \int_{x_j=x_{j-1}}^{t_j} \int_{x_{j+1}=x_j}^{t_j} \cdots \int_{x_k=x_{k-1}}^{t_j} dx_1 \ldots dx_k \tag{5}$$

and

$$f_{j,k} = F_k(t_j) = \int_{x_k=0}^{t_j} \int_{x_{k+1}=x_k}^{t_{k+1}} \cdots \int_{x_n=x_{n-1}}^{t_n} dx_k \ldots dx_n. \tag{6}$$

The first four steps of the following algorithm will compute these values in $O(n^3)$ and the next two steps will use these values to compute transient queue length and waiting time estimates. The validity of the algorithm will be demonstrated in Section 2.3.

## 2.2 An exact $O(n^3)$ Algorithm

**STEP 0** (Initialization).

Let

$$h_{0,0} = 1, f_{n+1,n+1} = 1, t_0 = 0 \tag{7}$$

**STEP 1** (Calculation of the diagonal elements $h_{k,k}$).

For $k = 1$ to $n$,

$$h_{k,k} = \sum_{i=1}^{k} (-1)^{k-i} \frac{t_i^{k-i+1}}{(k-i+1)!} h_{i-1,i-1}. \tag{8}$$

**STEP 2** (Calculation of $h_{j,k}$).

For $j = 1$ to $n-1$

For $k = j+1$ to $n$,

$$h_{j,k} = \frac{t_j^k}{k!} - \sum_{i=1}^{j} \frac{(t_j - t_i)^{k-i+1}}{(k-i+1)!} h_{i-1,i-1}. \tag{9}$$

**STEP 3** (Calculation of the diagonal elements $f_{k,k}$).

For $k = n$ to $1$,

$$f_{k,k} = \sum_{i=k}^{n} (-1)^{i-k} \frac{t_k^{i-k+1}}{(i-k+1)!} f_{i+1,i+1}. \tag{10}$$

**STEP 4** (Calculation of $f_{j,k}$).

For $j = 1$ to $n-1$

For $k = j+1$ to $n$,

$$f_{j,k} = \sum_{i=k}^{n} (-1)^{i-k} \frac{t_j^{i-k+1}}{(i-k+1)!} f_{i+1,i+1}. \tag{11}$$

**STEP 5** (Estimates of transient queue length).

1. For $j = 1$ to $n-1$

$$E[Q(t_j)|O'(\vec{t},n)] = \frac{\sum_{k=j}^{n} k h_{j,k}[f_{k+1,k+1} - f_{j,k+1}]}{h_{n,n}} - j. \tag{12}$$

2. Given $t$, $t_j \leq t < t_{j+1}$

$$E[Q(t)|O'(\vec{t},n)] = \theta E[Q(t_{j+1})|O'(\vec{t},n)] + (1-\theta) E[Q(t_j)|O'(\vec{t},n)] \tag{13}$$

where $\theta = \frac{t-t_j}{t_{j+1}-t_j}$ and for $0 \leq k \leq n-1-j$

$$Pr\{Q(t) = k|O'(\vec{t},n)\} = \frac{H_{j,k+j}(t)[F_{k+j+1}(t_{k+j+1}) - F_{k+j+1}(t)]}{h_{n,n}} \tag{14}$$

and

$$Pr\{Q(t_n) = 0|O'(\vec{t}, n)\} = 1 \tag{15}$$

where

$$H_{j,k}(y) = \frac{y^k}{k!} - \sum_{i=1}^{j} \frac{(y - t_i)^{k-i+1}}{(k - i + 1)!} h_{i-1,i-1}, \tag{16}$$

and

$$F_k(y) = \sum_{i=k}^{n} \frac{(-1)^{i-k} y^{i-k+1}}{(i - k + 1)!} f_{i+1,i+1}. \tag{17}$$

where $h_{0,0}$ and $f_{n+1,n+1}$ are defined in (7).

**STEP 6** (Estimates of transient waiting time).

For $t_j \leq t_k - t < t_{j+1}$

$$Pr\{W_k \leq t|O'(\vec{t}, n)\} = \frac{\sum_{r=1}^{k} H_{j,r}(t_k - t)[F_{r+1}(t_{r+1}) - F_{r+1}(t_k - t)]}{h_{n,n}}, \tag{18}$$

where $H_{j,k}(y)$ and $F_k(y)$ are defined in (16) and (17), respectively.

From the first four steps of the algorithm all of the $h_{j,k}$'s and $f_{j,k}$ 's can be found in $O(n^3)$. From (13), to compute $E[Q(t)|O'(\vec{t}, n)]$ for only one value of $t$ can be done in $O(n^2)$ because not all $h_{j,k}$'s and $f_{j,k}$'s are required. However, from (13), to compute $E[Q(t)|O'(\vec{t}, n)]$ for all values of $t$ requires $O(n^3)$ with the bottleneck steps being STEP 2 and STEP 4. To speed the implementation of the algorithm, for $j = 1, 2, \ldots, n$, $j!$ should be computed once and stored as a vector. Also, note that the $f_{j,k}$'s and $h_{j,k}$'s need not be stored but instead can be computed and then immediately used in equation (12) to reduce the storage requirements of the algorithm to $O(n)$.

## 2.3 Proof of Correctness

In this subsection we prove that indeed the algorithm correctly computes the distributions of the queue length and waiting time. A basic ingredient of the analysis is the following lemma that simplifies the multidimensional integrals:

**Lemma 1**

$$\int_{x_1=z}^{y}\int_{x_2=x_1}^{y}\cdots\int_{x_k=x_{k-1}}^{y} dx_1\ldots dx_k = \frac{(y-z)^k}{k!}. \tag{19}$$

**Proof**: (by Induction).

For $k = 1$ (19) is trivial. Suppose (19) were true for $r = k$. Then, from the induction hypothesis

$$\int_{x_1=z}^{y}\int_{x_2=x_1}^{y}\cdots\int_{x_{k+1}=x_k}^{y} dx_1\ldots dx_{k+1} = \int_{x_1=z}^{y}\frac{(y-x_1)^k}{k!}dx_1 = \frac{(y-z)^{k+1}}{(k+1)!}.$$

Hence, (19) is true for r=k+1 and the Lemma follows by induction. □

Remark: The proof of Lemma 1 can be generalized to show that

$$\int_{x_1=z}^{y}\int_{x_2=x_1}^{y}\cdots\int_{x_k=x_{k-1}}^{y} f(x_k)dx_1\ldots dx_k = \int_{x_1=z}^{y}\frac{(y-x_1)^{k-1}}{(k-1)!}f(x_1)dx_1. \tag{20}$$

In the following proposition we justify steps 1 and 2 of the algorithm.

**Proposition 2** *STEP 1 and STEP 2 of the algorithm are implied by the definition of $h_{j,k}$. Also $H_{j,k}(y)$ defined in (3) satisfy (16).*

**Proof**: If we apply lemma 1 to (3) we obtain

$$H_{j,k}(y) = \int_{x_1=0}^{t_1}\int_{x_2=x_1}^{t_2}\cdots\int_{x_j=x_{j-1}}^{t_j}\frac{(y-x_j)^{k-j}}{(k-j)!}dx_1\ldots dx_j, \tag{21}$$

which, after performing the innermost integral, is equal to

$$\int_{x_1=0}^{t_1}\int_{x_2=x_1}^{t_2}\cdots\int_{x_{j-1}=x_{j-2}}^{t_{j-1}}[\frac{(y-x_{j-1})^{k-j+1}}{(k-j+1)!} - \frac{(y-t_j)^{k-j+1}}{(k-j+1)!}]dx_1\ldots dx_{j-1}.$$

Hence, from (3) and (21) we obtain

$$H_{j,k}(y) = H_{j-1,k}(y) - \frac{(y-t_j)^{k-j+1}}{(k-j+1)!}h_{j-1,j-1}.$$

Successive substitution of this equation gives,

$$H_{j,k}(y) = H_{1,k}(y) - \sum_{i=2}^{j}\frac{(y-t_i)^{k-i+1}}{(k-i+1)!}h_{i-1,i-1}.$$

But from (3)

$$H_{1,k}(y) = \int_{x_1=0}^{t_1} \frac{(y-x_1)^{k-1}}{(k-1)!} dx_1 = \frac{y^k}{k!} - \frac{(y-t_1)^k}{k!}$$

and thus (16) follows. Since $h_{j,k} = H_{j,k}(t_j)$, (9) is obtained. Moreover, from (5) and (21), $h_{k,k} = H_{k,k}(0)$ and thus we obtain (8). □

We now turn our attention to the justification of steps 3 and 4 of the algorithm.

**Proposition 3** *STEP 3 and STEP 4 of the algorithm are implied by the definition of $f_{j,k}$ in (6). Also $F_k(y)$ defined in (4) satisfy (17).*

**Proof:** From (4) we note that $F_k(y)$ is a polynomial in $y$ of degree $n-k+1$. Let $c_{i,k}$ be the coefficients in $F_k(y) = \sum_{i=k}^n c_{k,i-k+1} y^{i-k+1}$. Moreover, from (4) we obtain:

$$F_k(y) = \int_{x_k=0}^{y} [F_{k+1}(t_{k+1}) - F_{k+1}(x_k)] dx_k.$$

By differentiating,

$$\frac{dF_k(y)}{dy} = F_{k+1}(t_{k+1}) - F_{k+1}(y) \tag{22}$$

and

$$\frac{d^2F_k(y)}{dy^2} = -\frac{dF_{k+1}(y)}{dy}.$$

More generally,

$$\frac{d^{i-k+1}F_k(y)}{dy^{i-k+1}} = (-1)^{i-k} \frac{dF_i(y)}{dy}.$$

From Taylor's Theorem, $c_{k,i-k+1} = \frac{d^{i-k+1}F_k(0)}{dy^{i-k+1}}/(i-k+1)!$ and $c_{i,1} = \frac{dF_i(0)}{dy}$. Therefore,

$$c_{k,i-k+1} = \frac{(-1)^{i-k}}{(i-k+1)!} c_{i,1}. \tag{23}$$

Evaluating (22) at $y = 0$ gives

$$c_{k,1} = \frac{dF_k(0)}{dy} = F_{k+1}(t_{k+1}),$$

since $F_{k+1}(0) = 0$. As a result,

$$c_{k-1,1} = F_k(t_k) = \sum_{i=k}^{n} c_{k,i-k+1} t_k^{i-k+1}.$$

From (23),

$$c_{k-1,1} = \sum_{i=k}^{n} \frac{(-1)^{i-k} t_k^{i-k+1}}{(i-k+1)!} c_{i,1},$$

where $c_{n,1} = 1$. Thus, $f_{k,k} = F_k(t_k) = c_{k-1,1}$ satisfy (10). In addition, $F_k(y) = \sum_{i=k}^{n} c_{k,i-k+1} y^{i-k+1}$, which from (23) is $\sum_{i=k}^{n} \frac{(-1)^{i-k} t^{i-k+1}}{(i-k+1)!} c_{i,1}$ and thus (17) follows. Moreover, since $f_{j,k} = F_k(t_j)$, (11) follows. □

Having established the validity of the steps of the algorithm required to compute the $h_{j,k}$'s and $f_{j,k}$'s we proceed to proof the correctness of the performance measure parts of the algorithm, steps 5 and 6.

**Theorem 4** *The distribution of $Q(t)$ and its mean conditioned on the observations $O'(\vec{t}, n)$ is given by (12), (13),(14) and (15) in STEP 5 of the algorithm.*

**Proof:** For $n - 1 \geq k \geq j$ and $t_j \leq t < t_{j+1}$ the event $\{N(t) = k, O(\vec{t})\}$ occurs if and only if $0 \leq X_1 \leq t_1, \ldots, X_{j-1} \leq X_j \leq t_j, X_j \leq X_{j+1} \leq t, \ldots, X_{k-1} \leq X_k \leq t, t \leq X_{k+1} \leq t_{k+1}, \ldots, X_{n-1} \leq X_n \leq t_n$. Since the arrival process is Poisson, the conditional density function is uniform and hence,

$$Pr\{N(t) = k, O(\vec{t}) | N(t_n) = n\} =$$

$$\frac{n!}{t_n^n} \int_{x_1=0}^{t_1} \cdots \int_{x_j=x_{j-1}}^{t_j} \int_{x_{j+1}=x_j}^{t} \cdots \int_{x_k=x_{k-1}}^{t} \int_{x_{k+1}=t}^{t_{k+1}} \int_{x_n=x_{n-1}}^{t_n} dx_1 \ldots dx_n,$$

which, from (3) and (4), is

$$\frac{n!}{t_n^n} H_{j,k}(t)[F_{k+1}(t_{k+1}) - F_{k+1}(t)].$$

Similarly, from (2) and (5),

$$Pr\{O(\vec{t}) | N(t_n) = n\} = \frac{n!}{t_n^n} \int_{x_1=0}^{t_1} \int_{x_2=x_1}^{t_2} \cdots \int_{x_n=x_{n-1}}^{t_n} dx_1 \ldots dx_n = \frac{n!}{t_n^n} h_{n,n}.$$

Therefore, for $n - 1 \geq k \geq j$ and $t_j \leq t < t_{j+1}$

$$Pr\{N(t) = k | O(\vec{t}), N(t_n) = n\} = \frac{H_{j,k}(t)[F_{k+1}(t_{k+1}) - F_{k+1}(t)]}{h_{n,n}}. \tag{24}$$

For $t = t_j$, using (5) and (6), we have

$$Pr\{N(t_j) = k | O(\vec{t}), N(t_n) = n\} = \frac{h_{j,k}[f_{k+1,k+1} - f_{j,k+1}]}{h_{n,n}}. \tag{25}$$

Since $Q(t) = N(t) - D(t)$ and $D(t_j) = j$

$$E[Q(t_j)|O(\vec{t}),\ N(t_n) = n] = E[N(t_j)|O(\vec{t}),\ N(t_n) = n] - j$$

$$= \sum_{k=j}^{n} k Pr\{N(t_j) = k | O(\vec{t}),\ N(t_n) = n\} - j$$

because, for $k < j$, $Pr\{N(t_j) = k|O(\vec{t}),\ N(t_n) = n\} = 0$. Hence, (12) follows from (25). Also, for $t_j \leq t < t_{i+j}$,

$$Pr\{Q(t) = k|O(\vec{t}),\ N(t_n) = n\} = Pr\{N(t) = j + k|O(\vec{t}),\ N(t_n) = n\}$$

so (14) follows from (24). In addition, (15) follows from the observation that $N(t_n) = D(t_n) = n$ with probability 1 and $Q(t) = N(t) - D(t)$.

Suppose we condition on the event $\{N(t_j) = r$ and $N(t_{j+1})\} = m$. Then, for $t_j \leq t < t_{j+1}$,

$$Pr\{N(t) = k|O(\vec{t}),\ N(t_n) = n,\ N(t_j) = r,\ N(t_{j+1}) = m\} =$$

$$\frac{Pr\{N(t) = k,\ N(t_j) = r,\ N(t_{j+1}) = m,\ O(\vec{t})\ |N(t_n) = n\}}{Pr\{N(t_j) = r,\ N(t_{j+1}) = m,\ O(\vec{t}),\ |N(t_n) = n\}} =$$

$$\frac{\int_{x_1=0}^{t_1} \cdot \int_{x_j=x_{j-1}}^{t_j} \cdot \int_{x_r=x_{r-1}}^{t_j} \int_{x_{r+1}=t_j}^{t} \cdot \int_{x_k=x_{k-1}}^{t} \int_{x_{k+1}=t}^{t_{j+1}} \cdot \int_{x_m=x_{m-1}}^{t_{j+1}} \int_{x_{m+1}=t_{j+1}}^{t_{m+1}} \cdot \int_{x_n=x_{n-1}}^{t_n} dx_1 . dx_n}{\int_{x_1=0}^{t_1} \cdot \int_{x_j=x_{j-1}}^{t_j} \cdot \int_{x_r=x_{r-1}}^{t_j} \int_{x_{r+1}=t_j}^{t_{j+1}} \cdot \int_{x_m=x_{m-1}}^{t_{j+1}} \int_{x_{m+1}=t_{j+1}}^{t_{m+1}} \cdot \int_{x_n=x_{n-1}}^{t_n} dx_1 . dx_n}$$

$$= \frac{\int_{x_{r+1}=t_j}^{t} \cdots \int_{x_k=x_{k-1}}^{t} \int_{x_{k+1}=t}^{t_{j+1}} \cdots \int_{x_m=x_{m-1}}^{t_{j+1}} dx_{r+1} \ldots dx_m}{\int_{x_{r+1}=t_j}^{t_{j+1}} \cdots \int_{x_m=x_{m-1}}^{t_{j+1}} dx_{r+1} \ldots dx_m}$$

after simplification. Using Lemma 1 and some additional simplification we obtain

$$Pr\{N(t) = k|O(\vec{t}), N(t_n) = n,\ N(t_j) = r,\ N(t_{j+1}) = m\} = \binom{m-r}{k-r} \theta^{k-r}(1-\theta)^{m-k}$$

where $\theta = \frac{t - t_j}{t_{j+1} - t_j}$. Hence,

$$Pr\{N(t) - N(t_j) = k'|O(\vec{t}),\ N(t_n) = n,\ N(t_j),\ N(t_{j+1})\}$$

$$= \binom{N(t_{j+1}) - N(t_j)}{k'} \theta^{k'} (1-\theta)^{N(t_{j+1})-N(t_j)} \tag{26}$$

Therefore, $N(t)$ conditioned on $N(t_j)$ and $N(t_{j+1})$ is a Binomial distribution so

$$E[N(t)|O(\vec{t}),\ N(t_n) = n,\ N(t_j),\ N(t_{j+1})] = N(t_j) + \theta(N(t_{j+1}) - N(t_j)),$$

which by taking expectations leads to

$$E[N(t)|O(\vec{t}),\ N(t_n) = n] = (1-\theta)E[N(t_j)] + \theta E[N(t_{j+1})]$$

Since $Q(t) = N(t)$ - j for $t_j \le t < t_{j+1}$ the piecewise linearity of (13) follows. □

Remark: Larson [1] also contains a proof of the piecewise linearity of $E[N(t)|O(\vec{t}), N(t_n) = n]$. Our proof more easily generalizes to the case of time varying arrival rates.

Finally, we establish the correctness of STEP 6 of the algorithm.

**Theorem 5** *The distribution of $W(t)$ conditioned on the observations $O'(\vec{t}, n)$ is given by (18) in STEP 6 of the algorithm.*

**Proof:** The total time $W_k$ the $k$th customer spent in the system is simply $t_k - X_k$. Therefore, the event $\{W_k \le t\}$ is the same as the event $\{X_k \ge t_k - t\}$. But this is the same as the event $\{N(t_k - t) \le k\}$. Hence,

$$Pr\{W_k \le t|O'(\vec{t}, n)\} = Pr\{N(t_k - t) \le k|O'(\vec{t}, n)\} = \sum_{i=1}^{k} Pr\{N(t_k - t) = i|O'(\vec{t}, n)\},$$

so (18) follows from (24). □

Note that (13) and (18) provides the basis to easily compute higher moments of $W(t)$, and $Q(t)$.

# 3 The Time-Varying Poisson Arrival Process

In the previous section we assumed that the arrival rates are Poisson with a hazard rate $\lambda$ that is constant during each busy period and found the inferred queue length and waiting time. In some applications, however, it is not realistic to assume that

$\lambda$ remains constant. We will show in this section that all the results of the previous section are immediately extendible to this case without adding any complexity to the resulting algorithm. In particular, if the arrival rate is time-varying, $\lambda(t) > 0$, then we will show that *the algorithm in Section 2.2 is applicable if all t's are replaced by* $\int_0^t \lambda(x)dx$.

Let $\lambda(t)$ be the time-varying arrival rate. We will first need to find the generalization of the conditional distribution of the order statistics given in (1).

**Theorem 6** *Given that $N(t) = n$ and the time-varying arrival rate $\lambda(t)$, the conditional density function of the n arrival times $X_1, X_2, \ldots, X_n$ is*

$$f(X_1 = x_1, \ldots, X_n = x_n | N(t) = n) = \frac{n! \prod_{i=1}^n \lambda(x_i)}{\Lambda^n(t)}, \tag{27}$$

*where* $\Lambda(t) = \int_0^t \lambda(x)dx$.

**Proof:** The proof below is similar to Theorem 2.3.1 of Ross [3]. Let $0 < x_1 < x_2 < \ldots x_n < x_{n+1} = t$, and let $\epsilon_i$ be small enough so that $x_i + \epsilon_i < x_{i+1}$, $i = 1, \ldots, n$. Now, $Pr\{x_1 \leq X_1 \leq x_1 + \epsilon_1, \ldots, x_n \leq X_n \leq x_n + \epsilon_n \text{ and } N(t) = n\} = Pr\{$exactly one arrival in $[x_i, x_i + \epsilon_i]$ for $i \leq n$ and no arrival elsewhere in $[0, t]\}$.

But the number of Poisson arrivals in an interval $I$ is Poisson with a mean $\int_{x \in I} \lambda(x)dx$, so the above probability is

$$\prod_{i=1}^n [e^{-M_i} M_i] e^{-(\Lambda(t) - \sum_{i=1}^n M_i)} = e^{-\Lambda(t)} \prod_{i=1}^n M_i$$

where $M_i = \int_{x_i}^{x_i + \epsilon_i} \lambda(x)dx = \lambda(x_i)\epsilon_i + o(\epsilon_i)$

Since $Pr\{N(t) = n\} = e^{-\Lambda(t)} \frac{\Lambda^n(t)}{n!}$ (27) follows.□

Given that we have now characterized the joint conditional density we can answer questions about the system behavior.

**Proposition 7** *Suppose $\lambda(t) > 0$. Let $t_j \leq t < t_{j+1}$. Then*

$$Pr\{X_1 \leq t_1, \ldots, X_n \leq t_n | N(t_n) = n\} = \frac{n! \int_{x_1=0}^{\Lambda(t_1)} \int_{x_2=x_1}^{\Lambda(t_2)} \cdots \int_{x_n=x_{n-1}}^{\Lambda(t_n)} dx_1 \ldots dx_n}{\Lambda(t_n)^n}. \tag{28}$$

**Proof**

From (27),

$$Pr\{X_1 \leq t_1, \ldots, X_n \leq t_n | N(t_n) = n\} = \frac{n! \int_{x_1=0}^{t_1} \int_{x_2=x_1}^{t_2} \cdots \int_{x_n=x_{n-1}}^{t_n} \prod_{i=1}^{n} \lambda(x_i) dx_1 \ldots dx_n}{\Lambda(t_n)^n}. \tag{29}$$

Consider now the transformation of variables $y_i = \Lambda(x_i)$, $i = 1, \ldots, n$. This is an one-to-one monotonically increasing function if $\lambda(t) > 0$. Furthermore $dy_i = \lambda(x_i)dx_i$. With this transformation of variables all of the upper limits change from $x_i = t_i$ to $y_i = \Lambda(t_i)$. But the condition $x_i = x_{i-1}$ is equivalent to $y_i = y_{i-1}$. Performing the transformation of variables we obtain (28).□

The proof of above proposition leads to a very simple extension of the algorithm of the previous section.

**Theorem 8** *STEPS 1-6 of the algorithm of Section 2.2 are valid for time varying $\lambda(t)$ if replace $t$ is replaced by $\Lambda(t)$ ( and $t_j$ with $\Lambda(t_j)$).*

**Proof**: This theorem follows immediately by applying the original proofs of STEPS 1-6 to the time varying case and using the transformation of variables $y_i = \Lambda(x_i)$ as was done in the above proposition. □

Remark: Note that the piecewise linearity property with respect to t of $E[Q(t)|O'(\vec{t}, n)]$ in equation (13) found for the case of constant $\lambda$ is destroyed by the transformation of variable, i.e., $E[N(t)|O'(\vec{t}, n)] = (1 - \theta)E[N(t_j)|O'(\vec{t}, n)] + \theta E[N(t_{j+1})|O'(\vec{t}, n)]$ where $\theta = \frac{\Lambda(t)-\Lambda(t_j)}{\Lambda(t_{j+1})-\Lambda(t_j)}$.

Note that while, the performance measures in the homogeneous Poisson case of the previous section do not depend on the knowledge of $\lambda$, the results in the time-varying case require knowledge of $\Lambda(t)$.

# 4 Stationary Renewal Arrival Process

Our methods in the previous sections critically depended on the Poisson assumption. In this section this assumption is relaxed and we only assume that the arrival process is a renewal time-homogeneous process, where the probability density function between two successive arrivals is a known function $f(x)$ and c.d.f. $F(x) = \int_{y=0}^{x} f(y)dy$. The key step is to generalize (2):

**Theorem 9**

$$Pr\{X_1 \leq t_1, \ldots, X_n \leq t_n | N(t_n) = n\} =$$
$$\frac{\int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} f(x_1)f(x_2-x_1)\ldots f(x_n-x_{n-1})(1-F(t_n-x_n))dx_1\ldots dx_n}{F^{(n)}(t_n) - F^{(n+1)}(t_n)}. \tag{30}$$

**Proof:** The underlying conditional density function can be expressed in terms of the ratio i.e.,

$$h(X_1 = x_1, \ldots, X_n = x_n | N(t) = n) = \frac{g\{X_1 = x_1, \ldots, X_n = x_n, N(t) = n\}}{Pr\{N(t) = n\}}.$$

But $g\{X_1 = x_1, \ldots, X_n = x_n, N(t) = n\}$ is the joint density function corresponding to the first interarrival time being $x_1$, the second interarrival time being $x_2 - x_1, \ldots$, the the $n$th interarrival being $x_n - x_{n-1}$,and having no arrival occurred in an interval of duration $t - x_n$. Hence,

$$g\{X_1 = x_1, \ldots, X_n = x_n, N(t) = n\} = f(x_1)f(x_2-x_1)\ldots f(x_n-x_{n-1})(1-F(t-x_n)).$$

Also,

$$Pr\{N(t) = n\} = Pr\{N(t) \leq n\} - Pr\{N(t) \leq n-1\} = Pr\{X_{n+1} \geq t)\} - Pr\{X_n \geq t)\}$$
$$= F^{(n)}(t) - F^{(n+1)}(t),$$

where $F^{(n)}(t)$is the cdf of the $n$th convolution of the renewal process and $F(x) = \int_0^x f(x)dx$ so

$$h(X_1 = x_1, \ldots, X_n = x_n | N(t) = n) = \frac{f(x_1)f(x_2-x_1)\ldots f(x_n-x_{n-1})(1-F(t-x_n))}{F^{(n)}(t) - F^{(n+1)}(t)} \tag{31}$$

so (30) follows. □

Equation (30) is the basis of the algorithm for estimating $Q(t)$ and $W(t)$.

For example, using (30), one can evaluate

$$Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\}$$

$$= \frac{Pr\{X_1 \leq t_1, \ldots, X_j \leq t_j, X_{j+1} \leq t_j, \ldots, X_k \leq t_j, X_{k+1} \leq t_{k+1}, \ldots, X_n \leq t_n | N(t_n) = n\}}{Pr\{X_1 \leq t_1, \ldots, X_j \leq t_j, X_{j+1} \leq t_{j+1}, \ldots, X_k \leq t_k, \ldots, X_{k+1} \leq t_{k+1}, \ldots, X_n \leq t_n | N(t_n) = n\}}. \tag{32}$$

From (32), one can find

$$E[Q(t_j) | O(\vec{t}), N(t_n) = n] = E[N(t_j) \geq k | O(\vec{t}), N(t_n) = n] - j$$

because $Q(t_j) = N(t_j) - j$.

$$= \sum_{k=1}^{n} Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} - j$$

$$= \sum_{k=j}^{n} Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} - 1$$

because $Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} = 1$ for $k = 1, 2, .., j-1$.

For the Erlang-k distribution, for example, $f(x) = \lambda(\lambda x)^{k-1} e^{-\lambda x}/(k-1)!$ and $F(x) = 1 - e^{-\lambda x} \sum_{j=0}^{k-1} (\lambda x)^j / j!$.

In this case, from (30), for *all* $t_1, t_2, \ldots, t_n$,

$$Pr\{X_1 \leq t_1, \ldots, X_n \leq t_n | N(t_n) = n\}$$

$$= A_k(\lambda, t_n) \int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} x_1^{k-1} (x_2 - x_1)^{k-1} \ldots (x_n - x_{n-1})^{k-1} \sum_{j=0}^{k-1} (\lambda(t_n - x_n))^j / j! dx_1 \ldots dx_n \tag{33}$$

where

$$A_k(\lambda, t_n) = \frac{(\lambda^k/(k-1)!)^n e^{-\lambda t_n}}{F^{(n)}(t_n) - F^{(n+1)}(t_n)}.$$

Therefore, the evaluation of (32) can ignore the contribution of $A_k(\lambda, t_n)$ because it will cancel in the numerator and denominator.

# 5 Realtime Estimates

In some applications it is desirable not to wait until the end of a busy period to estimate the queue length. For example, if there is a possibility of realtime control of the service time, knowledge that the queue length *was* excessively large but it is currently zero is not of value. Instead a current estimate is needed. The analysis of this problem will be derived for the case of a time-varying Poisson process. However, for the case of a constant arrival rate the method can be easily specialized.

Suppose that immediately after the $n$th departure (time $t_n^+$) we observe exactly $n$ departures at times $t_i$, $i = 1, 2, \ldots, n$ during a busy period and the busy period *did not* end at time $t_n$ (as inferred from the commencement of a new service initiation at time $t_n$). We are interested in having an estimating the state of the system at time $t \geq t_n$ *without* any observation of $D(t)$ between time $t_n$ and time $t$. Also note that knowledge that the busy period did not end at time $t_n$ is equivalent to $X_{n+1} \leq t_n$. The following theorem provides the basis for an O(n) on line algorithm for estimating $N(t)$ and $Q(t_n)$ given these observations.

**Theorem 10** *Let $t_n$ be the time we observe the system. Then, for $t > t_n$,*

$$E[N(t)|O(\vec{t}),\ X_{n+1} \leq t_n] = 1 + n + \Lambda(t) - \frac{R_n}{T_n} \tag{34}$$

*and*

$$E[Q(t_n)|O(\vec{t}),\ X_{n+1} \leq t_n] = 1 + \Lambda(t_n) - \frac{R_n}{T_n} \tag{35}$$

*where*

$$R_n = q_n + p_n - [\Lambda(t_n) + 1]e^{-\Lambda(t_n)} g_n, \tag{36}$$

$$T_n = p_n - e^{-\Lambda(t_n)} g_n, \tag{37}$$

*and $q_n, p_n, g_n$ are computed as follows:*

$$q_n = q_{n-1} + p_{n-1} - [\Lambda(t_n) + 1]e^{-\Lambda(t_n)} g_{n-1}, \tag{38}$$

$$p_n = p_{n-1} - e^{-\Lambda(t_n)} g_{n-1}, \tag{39}$$

*and*

$$g_k = \sum_{i=1}^{k} (-1)^{k-i} \frac{\Lambda^{k-i+1}(t_i)}{(k-i+1)!} g_{i-1} \tag{40}$$

*with the initial conditions* $p_0 = 1, q_0 = 0, g_0 = 1$.

**Proof:** For $j \geq n+1$, using (27),

$$Pr\{O(\vec{t}),\ X_{n+1} \leq t_n | N(t) = j\}$$

$$= \frac{j!}{\Lambda^j(t)} \int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} \int_{x_{n+1}=x_n}^{t_n} \cdots \int_{x_j=x_{j-1}}^{t} \prod_{i=1}^{j} \lambda(x_i) dx_i$$

$$= \frac{j!}{\Lambda^j(t)} \int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} \int_{x_{n+1}=x_n}^{t_n} \frac{[\Lambda(t) - \Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!} \prod_{i=1}^{n+1} \lambda(x_i) dx_i \tag{41}$$

from Lemma 1 and the transformation of variable $y_i = \Lambda(x_i), i = n+2, \ldots, j$. Also

$$Pr\{N(t) = j\} = e^{-\Lambda(t)} \frac{\Lambda^j(t)}{j!}. \tag{42}$$

Now, let

$$T_n = Pr\{O(\vec{t}), X_{n+1} \leq t_n\} = \sum_{j=n+1}^{\infty} Pr\{O(\vec{t}), X_{n+1} \leq t_n | N(t) = j\} Pr\{N(t) = j\}$$

$$= \sum_{j=n+1}^{\infty} e^{-\Lambda(t)} \int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} \int_{x_{n+1}=x_n}^{t_n} \frac{[\Lambda(t) - \Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!} \prod_{i=1}^{n+1} \lambda(x_i) dx_i \tag{43}$$

from (41) and (42), which leads to

$$T_n = \int_{x_1=0}^{t_1} \cdots \int_{x_n=x_{n-1}}^{t_n} \int_{x_{n+1}=x_n}^{t_n} e^{-\Lambda(x_{n+1})} \prod_{i=1}^{n+1} \lambda(x_i) dx_i. \tag{44}$$

But, for $t > t_n$,

$$E[N(t) | O(\vec{t}),\ X_{n+1} \leq t_n] = \sum_{j=n+1}^{\infty} j Pr\{N(t) = j | O(\vec{t}),\ X_{n+1} \leq t_n\}$$

$$= \sum_{j=n+1}^{\infty} (n+1+(j-n-1)) \frac{Pr\{O(\vec{t}),\ X_{n+1} \leq t_n | N(t) = j\} Pr\{N(t) = j\}}{Pr\{O(\vec{t}),\ X_{n+1} \leq t_n\}}.$$

From (41), (42) and (44) we find that this is equal to

$$\frac{1}{T_n}\sum_{j=n+1}^{\infty}(n+1+(j-n-1))e^{-\Lambda(t)}\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}\int_{x_{n+1}=x_n}^{t_n}\frac{[\Lambda(t)-\Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!}\prod_{i=1}^{n+1}\lambda(x_i)dx_i$$

$$=\frac{1}{T_n}[(n+1)T_n+\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}\int_{x_{n+1}=x_n}^{t_n}[\Lambda(t)-\Lambda(x_{n+1})]e^{-\Lambda(x_{n+1})}\prod_{i=1}^{n+1}\lambda(x_i)dx_i].$$

Therefore, (34) follows where

$$R_n=\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}\int_{x_{n+1}=x_n}^{t_n}\Lambda(x_{n+1})e^{-\Lambda(x_{n+1})}\prod_{i=1}^{n+1}\lambda(x_i)dx_i.$$

But, after computing the innermost integral using integration by parts,

$$R_n=\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}(-[\Lambda(t_n)+1]e^{-\Lambda(t_n)}+[\Lambda(x_n)+1]e^{-\Lambda(x_n)})\prod_{i=1}^{n}\lambda(x_i)dx_i$$

so (36) follows where

$$q_n=\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}\Lambda(x_n)e^{-\Lambda(x_n)}\prod_{i=1}^{n}\lambda(x_i)dx_i,$$

$$p_n=\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}e^{-\Lambda(x_n)}\prod_{i=1}^{n}\lambda(x_i)dx_i,$$

$$g_n=\int_{x_1=0}^{t_1}\cdots\int_{x_n=x_{n-1}}^{t_n}\prod_{i=1}^{n}\lambda(x_i)dx_i. \tag{45}$$

Computing the innermost integral of (44), we get (37). Integrating by parts we can find recursions from which the quantities $p_n, q_n, g_n$ are computed as follows:

$$q_n=\int_{x_1=0}^{t_1}\cdots\int_{x_{n-1}=x_{n-2}}^{t_{n-1}}(-[\Lambda(t_n)+1]e^{-\Lambda(t_n)}+[\Lambda(x_{n-1})+1]e^{-\Lambda(x_{n-1})})\prod_{i=1}^{n-1}\lambda(x_i)dx_1\ldots dx_{n-1}$$

so (38) follows.

Similarly,we get (39) from

$$p_n=\int_{x_1=0}^{t_1}\cdots\int_{x_{n-1}=x_{n-2}}^{t_{n-1}}[e^{-\Lambda(x_{n-1})}-e^{-\Lambda(t_n)}]\prod_{i=1}^{n-1}\lambda(x_i)dx_1\ldots dx_{n-1}.$$

From (5) and (45) and the transformation of variable, $y_i=\Lambda(x_i)$ described in Section 3, one finds that the definition of $g_n$ would equal $h_{n,n}$ if $t_i$ is replaced by $\Lambda(t_i)$. Therefore, from (8), $g_n$ satisfies (40).

Finally, $D(t_n) = n$ and $Q(t) = N(t) - D(t)$. Hence, (35) follows from (34). □

Remark: To find $E[Q(t)|O(\vec{t}),\ X_{n+1} \leq t_n]$ for $t \geq t_n$ requires knowledge of the departure process,i.e., knowledge of the service distribution and the number of servers present.

Theorem 10 provides an algorithm to update dynamically the queue length estimates after each departure. The algorithm is as follows:

**On Line Updating Algorithm**

**STEP 0** (Initialization).

Let $g_0 = 1$; $q_0 = 0$; $p_0 = 1$; $t_0 = 0$.

**STEP 1** (On line recursive update).

$$p_n = p_{n-1} - e^{-\Lambda(t_n)} g_{n-1},$$

$$q_n = q_{n-1} + p_{n-1} - [\Lambda(t_n) + 1]e^{-\Lambda(t_n)} g_{n-1},$$

$$g_n = \sum_{i=1}^{n} (-1)^{n-i} \frac{\Lambda^{n-i+1}(t_i)}{(n-i+1)!} g_{i-1},$$

$$R_n = q_n + p_n - [\Lambda(t_n) + 1]e^{-\Lambda(t_n)} g_n,$$

$$T_n = p_n - e^{-\Lambda(t_n)} g_n.$$

**STEP 2** (Real time estimation)

$$E[Q(t_n)|O(\vec{t}),\ X_{n+1} \leq t_n] = 1 + \Lambda(t) - \frac{R_n}{T_n}$$

and,for $t \geq t_n$,

$$E[N(t)|O(\vec{t}),\ X_{n+1} \leq t_n] = 1 + n + \Lambda(t) - \frac{R_n}{T_n}$$

At each step we need to keep track the previous two values $p_n, q_n$ and the vector $g_i$, $i = 1, \ldots, n$. The calculation of $p_n, q_n$ can be done in $O(1)$, since we can update the estimates for queue lengths in constant time given $p_{n-1}, q_{n-1}, g_{n-1}$. The calculation of $g_n$, however, requires $O(n)$ time. Using the same methodology we can recursively estimate the variance and higher moments of the queue length as well.

## 6 Computational Results

We implemented the algorithm of Section 2.2 for the case of stationary Poisson arrivals in a SUN 3 workstation. Using a straightforward implementation, the algorithm uses $n^2$ memory (the matrices $f_{j,k}$ and $h_{j,k}$ are half full). For most practical applications, the number of departures in a busy period is less than 100. We were able to calculate several performance characteristics with up to $n = 99$ number of departures in a busy period reliably in less than 40 seconds. In Figure 1 we report the expected number of arrivals in a busy period of 99 points given that the departure epochs are regularly spaced within the busy period, i.e. $t_i = i/n$. In Figure 2 we compute the estimate of the average cumulative number of arrivals during the interval parameterized by $n$, the total number of arrivals during the interval, based on the assumption that the departures are regularly spaced, i.e., given n we computed

$$\int_{t=0}^{t_n} E[Q(t)|\ O'(\vec{t}, n)]dt$$

Not surprisingly, this is a monotonically increasing function of $n$.

For the case of $n = 5$, we plot in Figure 3 the estimate of the queue length, $E[Q(t)|\ O'(\vec{t}, n)]$ as a function of time based on the assumption that the departures are at $t_i = i/5$, $\lambda = 10$, and the interarrival time is either exponentially distributed or Erlang-2 distributed. As expected the larger coefficient of variation of the exponential distribution causes a larger queue length. The exponential distribution curve is based on the algorithm in Section 2 where it was demonstrated that the curve is piecewise linear. The Erlang-2 distribution curve was based on the analysis in Section 4 using *Mathematica*. For this case piecewise linearity has not been established (indeed it does not hold) so the connecting line segments between the points $t_i$ should be viewed as an approximation. Note also that for exponential case the curve is convex as anticipated in Larson [1]. On the other hand, the Erlang-2 curve is clearly not convex.

If all of the parameters of Figure 3 remain unchanged except that $\lambda$ is increased to 100 then the exponential case curve would not change. This is because the algorithm in Section 2 is independent of $\lambda$. Figure 4 compares $\lambda = 10$ with $\lambda = 100$ for the case of Erlang-2 interarrival times. Not surprisingly, the estimates are quite close.

**Acknowledgments:**

The research of the first author was partially supported by the International Financial Services Center and the Leaders of Manufacturing program at MIT. The research of the second author was partially supported by NSF ECES-88-15449 and U.S. Army DAAL-03-83-K-0171 as well as the Laboratory for Information and Decision Sciences (LIDS) at MIT.

## References

[1] R. Larson (1990), "The Queue Inference Engine: Deducing Queue Statistics from Transactional Data", to appear in Management Science.

[2] R. Larson, (1989), "The Queue Inference Engine (QIE)", CORS/TIMS/ORSA Joint National Meeting, Vancouver,Canada.

[3] S. Ross (1983), *Stochastic Processes*, John Wiley, New York.

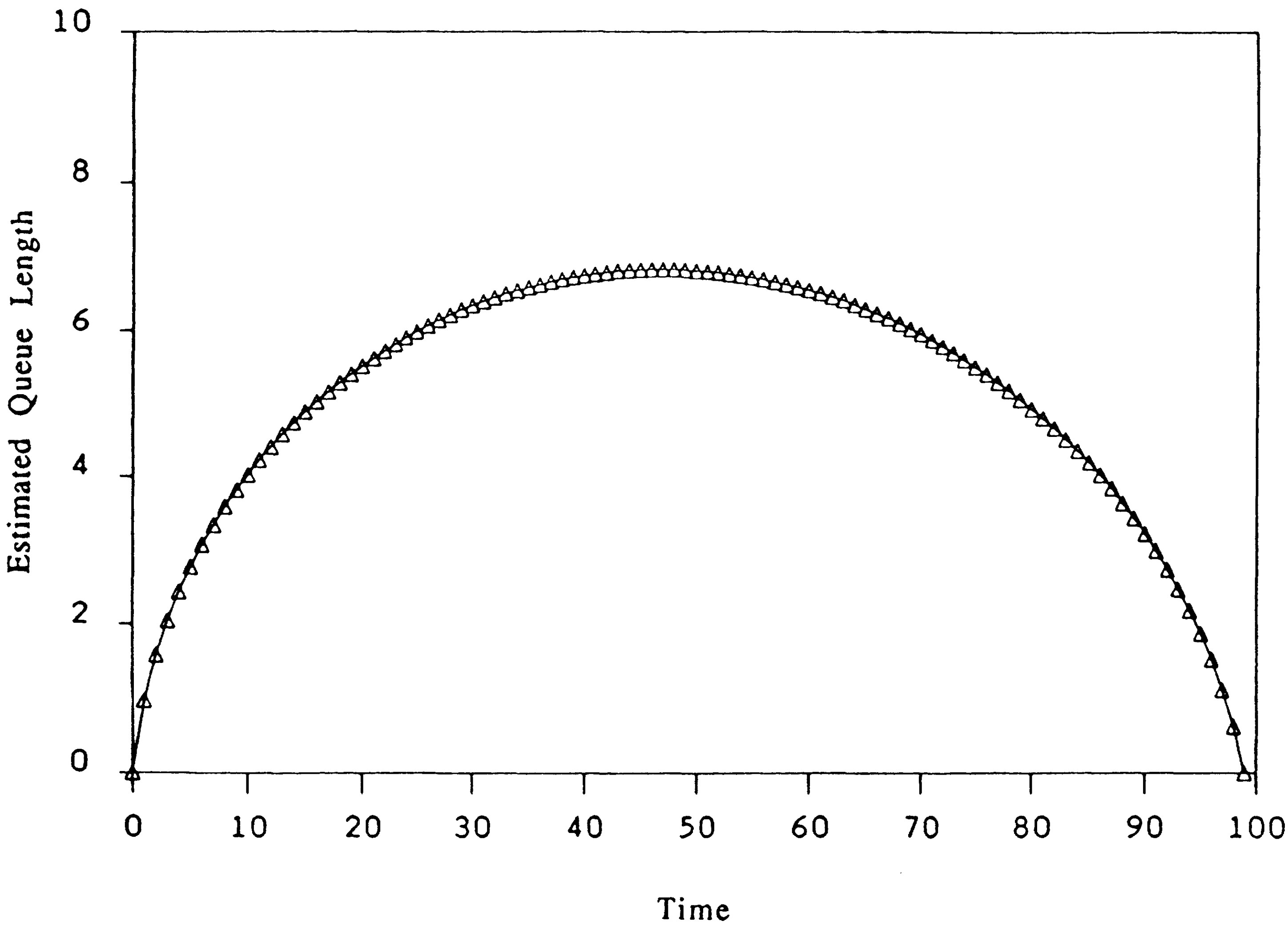


Figure 1: Estimated Queue Length vs Time
$n=99$, $t_i=i$.

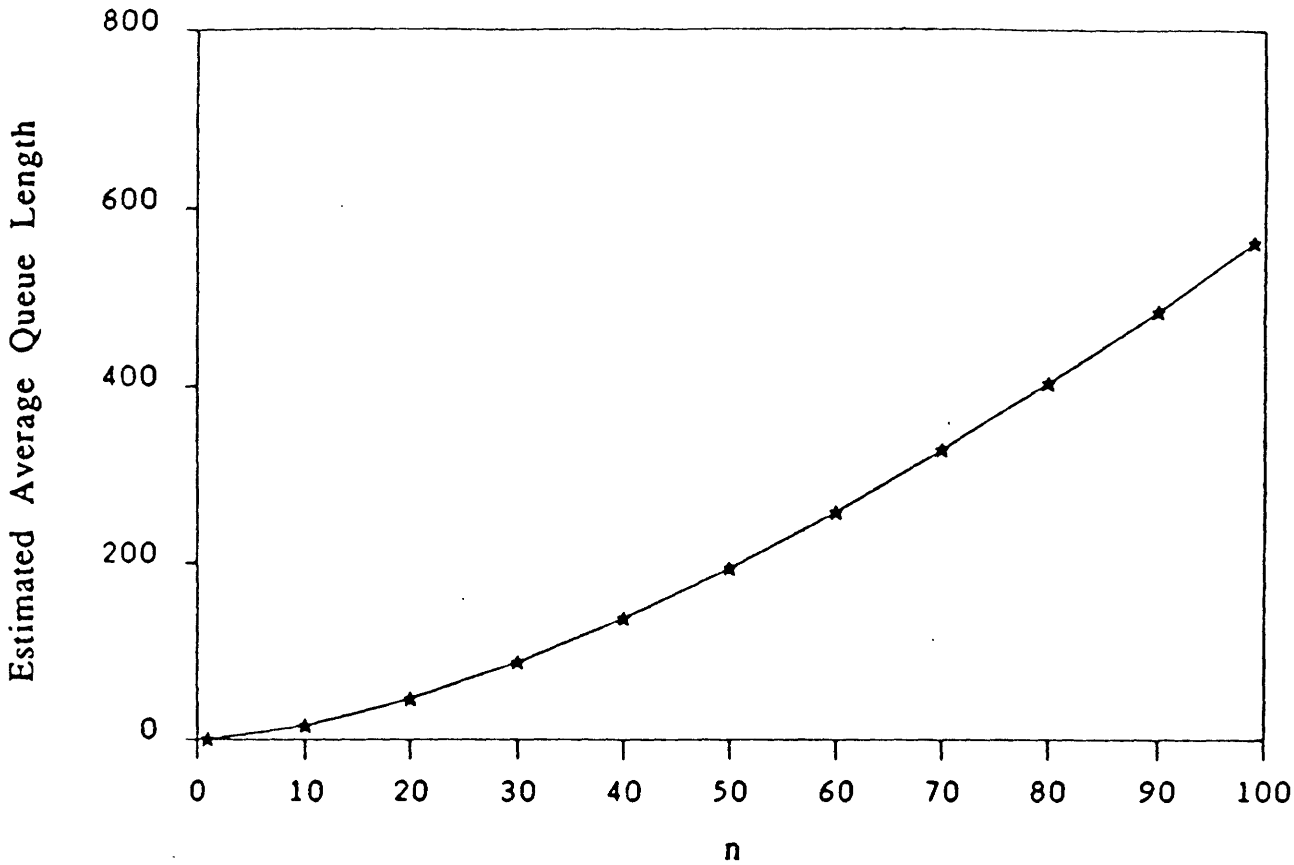


Figure 2: Estimated Average Queue Length vs n
$t_i = i/n$, n=1, 2,...,99.

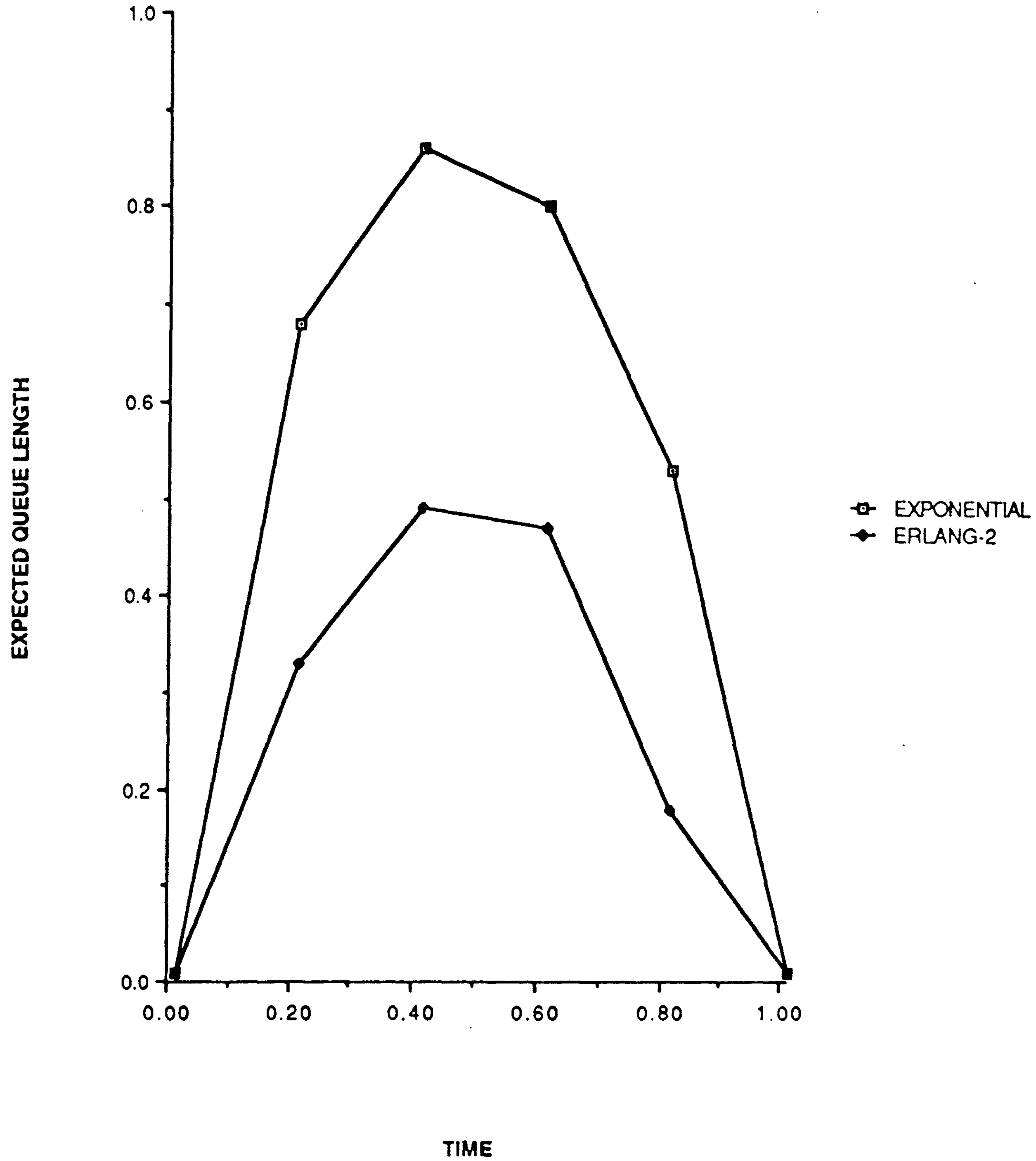


Figure 3: Expected Queue Length vs Time
Interarrival Distribution: Exponential or Erlang-2,
$n=5$, $t_i=i/5$, $\lambda=10$.

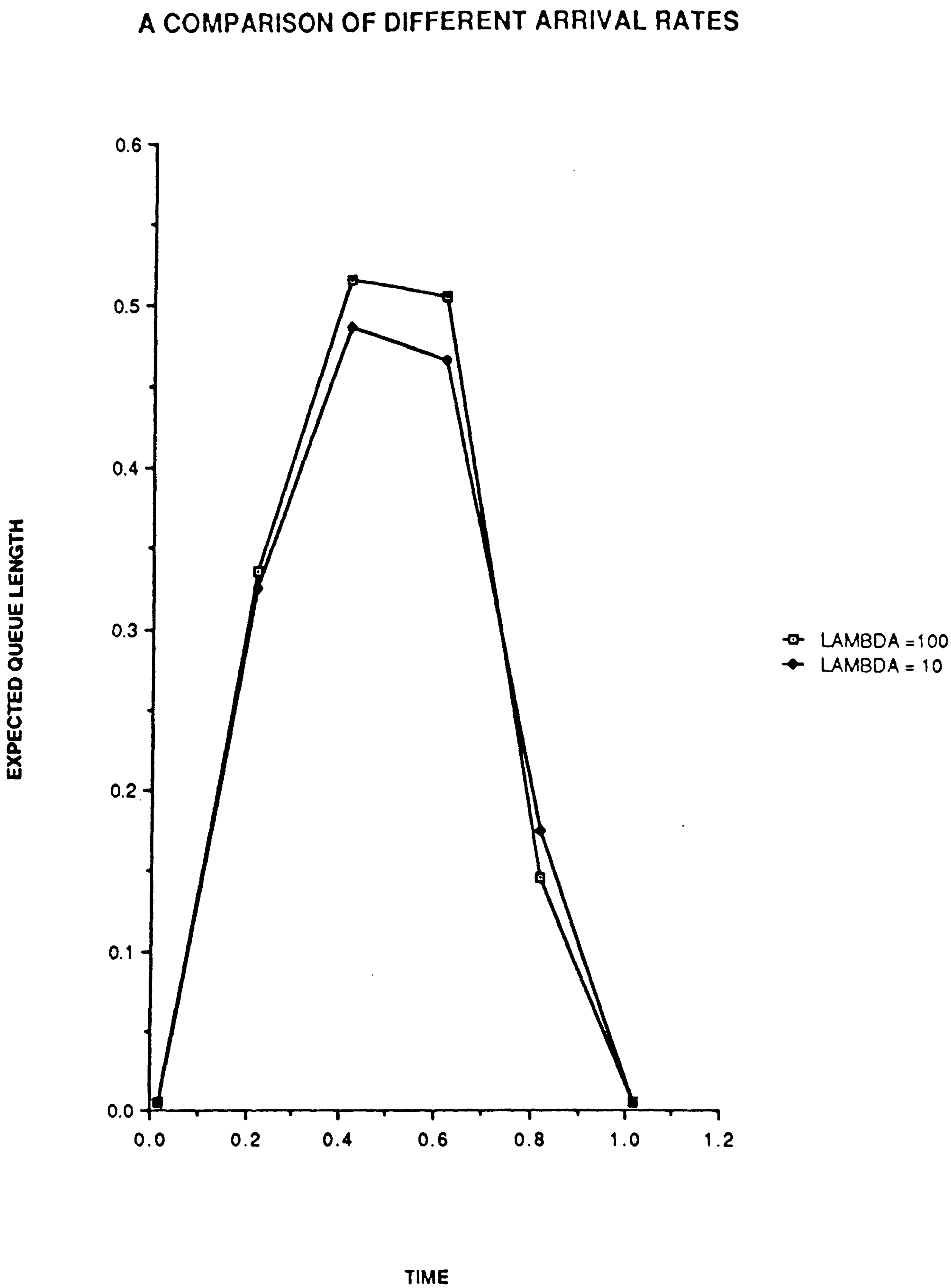


Figure 4: Expected Queue Length vs Time
Interarrival Distribution: Erlang-2,
$n=5$, $t_i=i/5$, $\lambda=10$ or 100.