Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

MODELING AND ANALYZING TRANSIENT MILITARY AIR TRAFFIC CONTROL

William Kaczynski

United States Military Academy
646 Swift Road
West Point, NY 10996, USA

Lawrence Leemis

The College of William & Mary
P.O. Box 8795
Williamsburg, VA 23187, USA

John Drew

The College of William & Mary
P.O. Box 8795
Williamsburg, VA 23187, USA

ABSTRACT

A theoretical application of transient queueing analysis is provided for military air traffic control. The exact distribution of the nth arriving or departing flight's sojourn time in an $M/M/s$ queue with k flights initially present is reviewed. Algorithms previously developed for computing the covariance between sojourn times for an $M/M/1$ queue with $k \geq 0$ flights present at time zero are provided and utilized. Maple computer code is utilized for practical applications in air traffic control of transient queue analysis for many system measures of performance without regard to traffic intensity (i.e., the system may be unstable with traffic intensity greater than one), thus negating the need for simulation.

1 INTRODUCTION

Many traditional simulation studies analyze queueing systems in steady-state, requiring appropriate warm-up periods and associated long simulation runs. However, in many cases the system being modeled never reaches steady-state; thus steady-state simulation results do not accurately portray the system behavior, as is often the case in military air traffic control. The ability to analyze transient results associated with such models is often complicated by intractable theory, leaving simulation as the only method for analysis. Further complicating the transient analysis is the effect of initial conditions (Kelton and Law, 1985). Since steady-state results depend on running the system long enough to mitigate the impact of initial conditions, these steady-state results reveal nothing about the transient behavior of the queueing system. Our purpose here is to combine new and existing results in transient queueing analysis with a symbolic engine in computational probability.

There are many classes of queueing systems where a transient analysis is required, e.g., military applications often use queueing models that never reach equilibrium. Recognizing the need to develop theory for transient results, as opposed to steady-state results, has resulted in a wide literature in this area. Initial work in transient analysis ironically appeared as an attempt to measure when a system achieved equilibrium. Law (1975) notes the consequences of failing to adequately account for the initial transient period, leading to Gafarian et al. (1976) outlining a comprehensive framework for the initial transient problem. Morisaku (1976) addresses the time to equilibrium in simulations modeling the $M/M/1$ queue and provides schematics for the transition probabilities given $k \geq 0$ customers initially present at time $t = 0$. Grassmann (1977) compares three methods for finding transient solutions in Markovian queueing systems, Runge–Kutta, Liou's method, and randomization, where randomization is shown as superior for large sparse transition matrices. Pegden and Rosenshine (1982) provide a closed-form solution for the probability of exactly i arrivals and j servicings over a time horizon of length t in an $M/M/1$ queue starting empty and idle, allowing the calculation of certain performance measures for a specified time period. Odoni and Roth (1983) take an empirical approach to compare observed and predicted transient state queue length for the $M/M/1$ queue, noting that for small values of t the expected queue length is strongly influenced by initial conditions, and provide a good approximation for an upper bound of time to steady-state. Kelton and Law (1985) consider the $M/M/s$ $(s \geq 1)$ queue and provide expressions for calculating the probabilities of having

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

up to $n+k$ customers in the system upon the arrival of the nth customer, where k is the number of customers in the system at time $t=0$. They then apply these calculations to a variety of measures of performance with implications to convergence on steady-state delays and offer methods for choosing queue initialization in simulation. Much of the work in this paper is motivated by their results. Kelton (1985) extends the previous work by considering $M/E_m/1$ and $E_m/M/1$ queues. Parthasarathy (1987) provides a transient solution for the probability that there are n customers in the system at time t for an $M/M/1$ queue. Abate and Whitt (1988) use Laplace transformations to analyze some transient results of interest in the $M/M/1$ queue. Leguesdron, et al. (1993) provide transient probabilities for the $M/M/1$ queue by inverting the generating function of the uniformized Markov chain describing the $M/M/1$ process. Transient distributions of cumulative reward are calculated by e Silva, et al. (1995) using uniformization, where the distribution of cumulative reward is over a finite interval with reward rates represented by Markov model states. Grassmann (2008) investigates warm-up periods in simulation and shows that in many cases these warm-up periods should not be used, especially if the simulation begins in a high probability state.

The purpose of this paper is to adapt previous work on transient analysis to air traffic control. Kaczynski et al. (2010) provide an extensive literature review in transient analysis. In this paper we will focus on the transient analysis of the $M/M/1$ and the more general $M/M/s$ queues as related to air traffic control, specifically on the distribution of the nth flight's (arriving or departing) sojourn time, which is the sum of the nth flight's delay time (referred to as holding time in aviation) and service time. Almost all the references listed in Kaczynski et al. (2010) address measures of performance specified over a finite time interval and are the results of numerical work, which is in stark contrast to the measures proposed here, which are based on the exact distribution of specific flights. Rather than arriving at the results numerically, a computer algebra system calculates exact measures of performance based on a given number of arriving and departing flights.

The $M/M/s$ queue is defined in Section 2 for a positive integer s, and a method is given for calculating the probability distribution of the number of aircraft an arriving or departing flight sees upon contact with air traffic control (ATC) (upon arrival to an $M/M/s$ queue). Section 3 describes how the sojourn time distribution is calculated for a given flight in an $M/M/s$ queue with k flights initially present in the system, $k \geq 0$. Section 4 includes examples using the implemented procedures to calculate exact sojourn time distributions, related measures of performance, and graphical illustrations for varying parameters such as traffic intensity and number of flights in the system. Section 5 applies two approaches for calculating the covariance and correlation among flights in an $M/M/1$ queue. Section 6 extends the covariance and correlation calculations by automating the process of finding the joint probability distribution function between two flights, and provides the exact covariance and correlation calculations for varying traffic intensities. Section 7 concludes the paper by reviewing the content and offering areas of further study. Commented code is available for all computations conducted here upon request.

2 BASICS OF THE $M/M/s$ QUEUE

The $M/M/s$ queue is governed by *iid* exponential interarrival times (the arrival stream is a Poisson process) with arrival rate λ, and *iid* exponential service times among s identical servers, each with service rate μ. The interarrival times and the service times are mutually independent. The traffic intensity of the system is $\rho = \lambda/s\mu$. The system consists of a single queue with customers waiting to be serviced by one of the identical s parallel servers. If an arriving customer finds at least one idle server, the customer immediately proceeds to service; otherwise the customer joins the single queue of those waiting for service in a first-come, first-served manner. To achieve classic steady-state results the traffic intensity must satisfy $\rho < 1$. This critical assumption is not required in transient analysis, described here, because the system of interest never reaches equilibrium. In this application, if $s=1$, we will assume there is a single controller and a single runway. If $s>1$, we will assume there is one runway for each controller and that the controllers have adequate space to independently serve flights.

Let $P_k(n,i)$ be the probability that upon the arrival of the nth flight there are exactly i flights in the system including the nth flight (in queue or in service), given k flights are present at time $t=0$. Using propositions provided by Kelton and Law (1985) and a recursion algorithm, $P_k(n,i)$ for $i=1,2,\ldots,n+k$ can be computed. Using these probabilities, it is possible to find the distribution of the sojourn time for the nth flight in an $M/M/s$ queue, given k flights are present at time $t=0$.

Kaczynski et al. (2010) includes these results coded in the Maple procedure `Queue(X, Y, n, k, s)`, where

- X is the exponential interarrival time distribution,
- Y is the exponential service time distribution,
- n is the index of the flight of interest,
- k is the number of flights in the system at time $t = 0$,
- s is the number of identical parallel servers.

The procedure is written in Maple and uses A Probability Programming Language (APPL), which can be downloaded for free at www.APPLsoftware.com and is described in Glen et al. (2001). We choose to calculate the distribution of the sojourn time because it is a purely continuous random variable enabling us to exploit associated procedures in APPL.

3 CREATING THE SOJOURN TIME DISTRIBUTION

Once the necessary $P_k(n,i)$, $i = 1,2,\ldots,n+k$, probabilities are calculated, the exact sojourn time distribution for the nth flight can be calculated. We define X_n as the number of flights, including flight n, in the system at time t, the arrival time of the nth flight. The possible values of X_n can vary from a minimum of 1, which occurs when flight n arrives to empty airspace, to a maximum of $n+k$, which occurs when 0 completions occur prior to flight n's arrival. The mathematical derivations for both the $M/M/1$ and $M/M/s$ queues make extensive use of the memoryless property, permitting the construction of the distribution of T_n, the sojourn time of flight n. We present each case separately below.

3.1 Distribution of T_n for the $M/M/1$ queue

Kaczynski et al. (2010) show the $M/M/1$ queue sojourn time distribution for $k = 0$ initial flights generalizes very elegantly to include $k > 0$, as indicated in Table 1. Line i of the table occurs with probability $P_k(n,i)$ and lists the distribution of the sojourn time for the nth flight, conditioned on i flights being in the system upon its arrival.

Table 1: Conditional sojourn time distributions for the $M/M/1$ queue.

X_n	Delay	Service	Conditional sojourn time distribution
1	0	exponential(μ)	exponential(μ)
2	exponential(μ)	exponential(μ)	Erlang(μ, 2)
3	Erlang(μ, 2)	exponential(μ)	Erlang(μ, 3)
4	Erlang(μ, 3)	exponential(μ)	Erlang(μ, 4)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n+k$	Erlang(μ, $n+k-1$)	exponential(μ)	Erlang(μ, $n+k$)

Let $g_i(t)$ be the PDF of an Erlang(μ, i) random variable. Using the conditional sojourn time distributions for $i = 1,2,\ldots,n+k$ potential flights in the system, each with probability $P_k(n,i)$, the probability density function (PDF) for the nth flight's sojourn time T_n is the mixture

$$f_n(t) = \sum_{i=1}^{n+k} P_k(n,i) g_i(t) \qquad t > 0. \tag{1}$$

3.2 Distribution of T_n for the $M/M/s$ queue

Given $s > 1$ parallel identical servers (controllers), the nth flight's sojourn time distribution is still a mixture of $n+k$ conditional sojourn time distributions. However, each distribution might be more complicated. For illustration, consider an $M/M/3$ queue starting empty and idle with exponential(λ) arrivals and three identical exponential(μ) servers. It is clear that for flights 1, 2, and 3, the sojourn time is exponential(μ) since all three flights proceed directly to service. Therefore, in the general case, for the number of flights in the system including flight n, which we defined as X_n, when $X_n \le s$ the conditional sojourn time distribution is exponential(μ). However, if $X_n > s$, then the nth flight experiences a delay while observing $X_n - s$ service completions. When $s > 1$ and $X_n > s$, the service distribution observed by flights in queue is exponential with rate $s\mu$. Using this result, it is apparent that the delay time for the nth flight is the sum of $X_n - s$ independent exponential($s\mu$) random variables, and using (1) is Erlang$(s\mu, X_n - s)$. To calculate the nth flight's sojourn time for a particular value of X_n, we sum the delay time and the service time. Table 2 shows the distributions conditioned on the number of flights X_n encountered by flight n (including itself) for the $M/M/3$ queue, given $k = 0$ flights present at time $t = 0$. The APPL procedure `Convolution` calculates the distribution of a sum of independent random variables. We use the symbol $\oplus$ to represent convolution.

Table 2: Conditional sojourn time distributions for the $M/M/3$ queue with $k = 0$.

X_n	Delay	Service	Conditional sojourn time distribution
1	0	exponential(μ)	exponential(μ)
2	0	exponential(μ)	exponential(μ)
3	0	exponential(μ)	exponential(μ)
4	exponential(3μ)	exponential(μ)	exponential(3μ) $\oplus$ exponential(μ)
5	Erlang(3μ, 2)	exponential(μ)	Erlang(3μ, 2) $\oplus$ exponential(μ)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	Erlang(3μ, $n-3$)	exponential(μ)	Erlang(3μ, $n-3$) $\oplus$ exponential(μ)

Since X_n represents the number of flights in the system upon arrival of the nth flight, including itself, the first row in Table 2 corresponds to flight n arriving to an empty system and the last row corresponds to no service completions prior to flight n's arrival. The general form for the $M/M/s$ sojourn time PDF is identical to (1), however, in the $M/M/s$ case each $g_i(t)$ can potentially require an additional step to calculate the distribution of a sum of random variables.

4 AIR TRAFFIC CONTROL TRANSIENT ANALYSIS APPLICATIONS

It is apparent that calculating the conditional sojourn time distribution for large n is tedious. Kelton and Law (1985) acknowledge the computational difficulty in achieving the $P_k(n,i)$ probabilities alone. Conducting the added steps of up to $n - s$ convolutions for the $M/M/s$ queue and then mixing the resulting conditional distributions with the appropriate probabilities can be complicated to implement. Due to these computational difficulties, simulation is often required for analysis, however, APPL provides the underlying computational engine to achieve exact results for such problems. As mentioned earlier, the APPL procedure `Queue(X, Y, n, k, s)` returns the exact sojourn time distribution for flight n. APPL is capable of symbolic results, as illustrated in Examples 1 and 2.

Example 1. Consider an $M/M/1$ queue (a single controller, single runway facility) with arrival rate λ and service rate μ starting empty and idle at time $t = 0$. For the fourth flight, calculate the probabilities $P_0(4,i)$ for $i = 1,2,3,4$.

The APPL command `MMsQprob(4, 0, 1)` returns exact symbolic probabilities, and for $\rho = 9/10$,

$$\begin{aligned}
P_0(4,1) &= \frac{5\rho^2+4\rho+1}{(\rho+1)^5} &= \frac{865000}{2476099} &\approx 0.3493\\
P_0(4,2) &= \frac{\rho\left(5\rho^2+4\rho+1\right)}{(\rho+1)^5} &= \frac{778500}{2476099} &\approx 0.3144\\
P_0(4,3) &= \frac{\rho^2(3\rho+1)}{(\rho+1)^4} &= \frac{29970}{130321} &\approx 0.2300\\
P_0(4,4) &= \frac{\rho^3}{(\rho+1)^3} &= \frac{729}{6859} &\approx 0.1063,
\end{aligned}$$

where $\rho = \lambda/\mu$. It is easy to verify that for any $\rho > 0$, $\sum_{i=1}^{4} P_0(4,i) = 1$, as required.

Example 2. For the queue described in Example 1, calculate the fourth flight's sojourn time distribution, mean sojourn time and sojourn time variance.

The APPL statements

```
X := ExponentialRV(lambda);
Y := ExponentialRV(mu);
T := Queue(X, Y, 4, 0, 1);
Mean(T);
Variance(T);
```

calculate the desired results. The first two lines define the interarrival and service time distributions, while the third line calculates the fourth flight's sojourn time distribution. The last two lines are self explanatory. The resulting PDF is

$$\begin{aligned} f_4(t) \quad &= \quad \mu^4 e^{-\mu t}\left(30\lambda^2+30\lambda^3 t+24\lambda\mu+24\lambda^2\mu t+6\mu^2+6\mu^2\lambda t+9t^2\lambda^4+\right. \\ &\quad \left. 12t^2\lambda^3\mu+3t^2\lambda^2\mu^2+t^3\lambda^5+2t^3\lambda^4\mu+t^3\lambda^3\mu^2\right)/6\left(\lambda+\mu\right)^5 \qquad t>0. \end{aligned}$$

Using $f_4(t)$ above, the `Mean` and `Variance` commands return

$$E\left[T_4\right]=\frac{\mu^5+6\lambda\mu^4+26\mu^2\lambda^3+16\mu^3\lambda^2+17\mu\lambda^4+4\lambda^5}{\mu\left(\lambda+\mu\right)^5}$$

and

$$\begin{aligned} V\left[T_4\right] \quad &= \quad \left(181\mu^2\lambda^8+484\mu^3\lambda^7+816\mu^4\lambda^6+868\mu^5\lambda^5+574\mu^6\lambda^4+\right. \\ &\quad \left. 244\mu^7\lambda^3+40\mu\lambda^9+68\mu^8\lambda^2+12\mu^9\lambda+\mu^{10}+4\lambda^{10}\right)/\left(\mu^2\left(\lambda+\mu\right)^{10}\right). \end{aligned}$$

Substituting $\lambda=1$ and $\mu=10/9$, the results simplify to

$$f_4(t) \quad = \quad \frac{5000}{66854673}e^{-10/9t}\left(361t^3+2109t^2+5190t+5190\right) \qquad t>0,$$

$$E\left[T_4\right]=\frac{23323347}{12380495}\approx 1.8839, \text{ and } V\left[T_4\right]=\frac{383506725720906}{153276656445025}\approx 2.5021.$$

The CPU time associated with the examples is negligible. Examples 1 and 2 represent simple applications of these procedures that circumvent time intensive hand-calculations. They serve only as indications of more challenging problems solvable using these procedures.

Example 3. Calculate the mean sojourn time of the 30th flight in an $M/M/2$ queue with arrival rate $\lambda=1$, service rate $\mu=9/20$ ($\rho=10/9$), and $k=3$ flights initially present.

The mean can be calculated in a single APPL statement by embedding the function calls

```
Mean(Queue(ExponentialRV(1),ExponentialRV(9/20),30,3,2));
```

which yields 9.634524585.

Representing the sojourn time distribution for the nth flight in closed form also provides valuable information on asymptotic behavior for queueing systems, including steady state convergence rates for different initial conditions. Figure 1 shows the mean sojourn time for flight $n=1,2,\ldots,120$ in an $M/M/1$ queue with $\lambda=1$, $\mu=10/9$, and $\rho=9/10$ for several values of k. The points that are plotted have been connected by lines. As expected, despite the initial condition, all cases appear to move toward the steady-state value of 9 with increasing n. The horizontal axis is only limited to $n=120$ for display purposes and in fact, identical computations were carried out for $n>300$ flights to verify convergence. However, as shown in the cases where $k=6$ and $k=10$, the convergence to steady-state is not always monotone. Additionally, in testing various traffic intensities, the rate of convergence to steady-state increases rapidly with decreasing traffic intensity for varying values of k.

APPL also has the ability to calculate the closed-form cumulative distribution function (CDF) for the nth flight's sojourn time permitting CDF comparisons for varying n as well as distribution

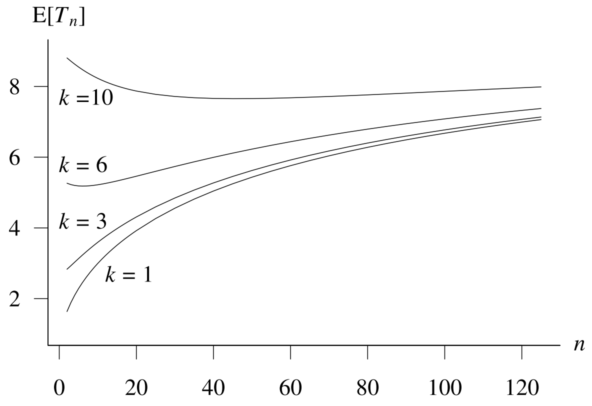

Figure 1: $M/M/1$ mean sojourn time for $\rho = 9/10$ given k at $t = 0$.

percentiles for a given flight. Varying k for an $M/M/1$ queue provides another basis for comparison of CDFs, where resulting CDFs are plotted across k allowing direct comparison of sojourn time percentiles for flight n.

Given the complete specification of the sojourn time distribution, one can use APPL to calculate not only the mean but also the 2nd, 3rd, and 4th moments for flight n. This is especially valuable for steady-state analysis. It is common in simulation to verify attainment of steady-state behavior by examining the mean delay or mean sojourn time. Though some literature exists on estimating transient mean and variance, we are not aware of any literature addressing higher moments. Literature addressing the second moment seems mostly focused on variance estimation and not necessarily convergence. Therefore, even when the first moment might acceptably approximate the steady state value, there is reason for further analysis of higher moments. For example, Figure 2 displays the first four moments of the sojourn time for flight n in an $M/M/1$ queue, where $\lambda = 1$, $\mu = 2$, $\rho = 1/2$, with the initial condition $k = 0, 4, 8$.

The vertical dashed lines give the smallest flight number for which all three of the transient values are within 1% of the steady state value. The relatively low traffic intensity $\rho = 1/2$ was selected purposely to allow quick convergence and easy visual inspection. Even with this somewhat low traffic intensity, it is apparent that the higher moments converge more slowly than the lower moments. In other scenarios where $\rho > 1/2$, the higher moments exhibit an even slower convergence. Each vertical dashed line in Figure 2 was triggered by the $k = 8$ curve, suggesting that the moments are more sensitive to a heavily pre-loaded system. For the cases $k = 0, 4, 8$, the flight numbers for which the transient results were within 1% of the steady state values are listed in Table 3. To verify the initial-condition effect on the convergence rate of the first four moments, k was increasingly incremented beyond eight and displayed a further slowing of convergence.

Table 3: Smallest flight number where the sojourn time transient result is within 1% of steady state for an $M/M/1$ queue with $k = 0, 4, 8$ and $\rho = 1/2$.

	$k = 0$	$k = 4$	$k = 8$
$E[T]$	19	21	36
$\sqrt{Var[T]}$	27	29	46
$E\left[((T-\mu)/\sigma)^3\right]$	28	29	50
$E\left[((T-\mu)/\sigma)^4\right]$	34	35	56

5 FLIGHT COVARIANCE AND CORRELATION IN THE $M/M/1$ QUEUE

The dependence exhibited in sojourn times of successive flights is one reason for the difficulty in calculating interval estimators for queue measures of performance. In the simplest case, consider an empty and idle $M/M/1$ queue with interarrival and service rates λ and μ. Our desire is to calculate the covariance between the sojourn times of any two flights for the first three flights of the day.

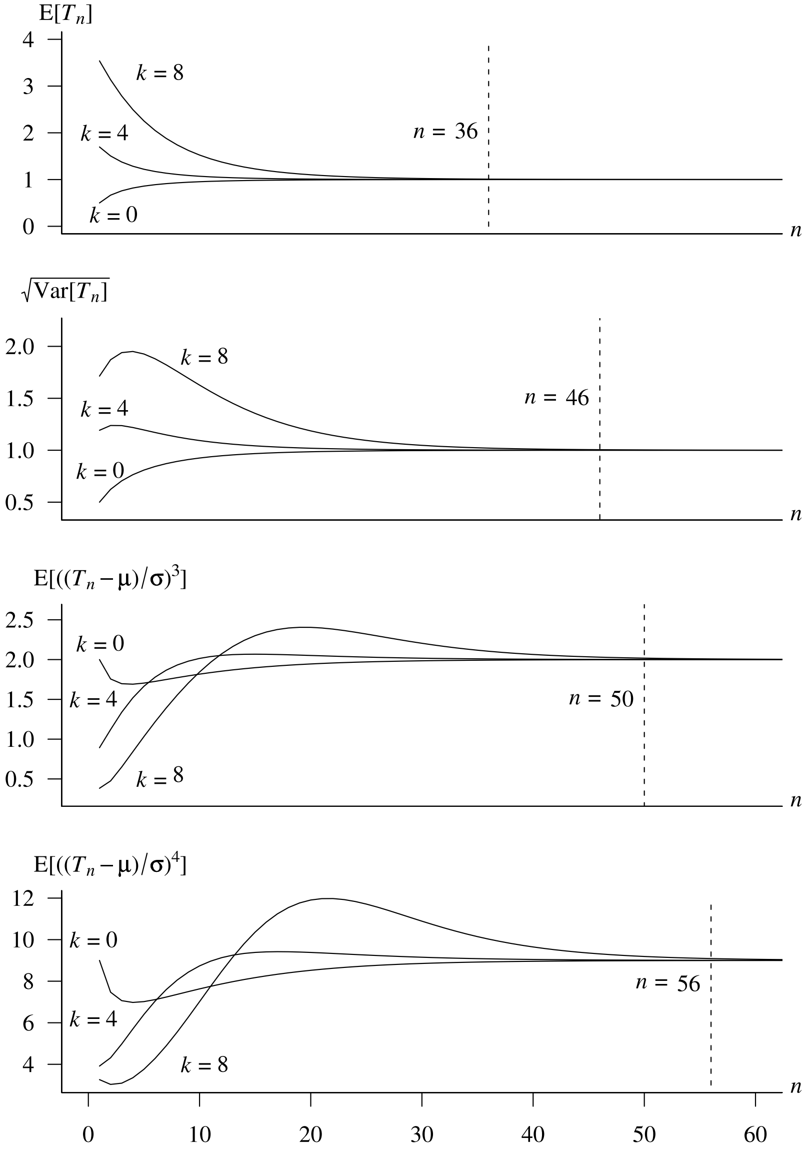

Figure 2: First four moments of the $M/M/1$ sojourn time for flights 2 through 60 for $\rho = 1/2$ and $k = 0, 4, 8$. The arrival rate is $\lambda = 1$ and the service rate is $\mu = 2$, resulting in steady-state values for the four measures of performance of $1, 1, 2, 9$.

When considering $n = 3$ flights, there are five possible ways flights can arrive and be serviced. In general, for n flights, the number of ways that flights can enter and leave the ATC is given by the nth Catalan number, which is

$$C_n = \frac{(2n)!}{(n!)(n+1)!}.$$

Figure 3 shows the five possible arrangements for $n = 3$ flights along with the sojourn times T_1, T_2, and T_3 for each, with the arrival and completion times for the ith flight denoted by a_i and c_i respectively. The vertical arrows at event times represent service completions (pointing up) or arrivals (pointing down).

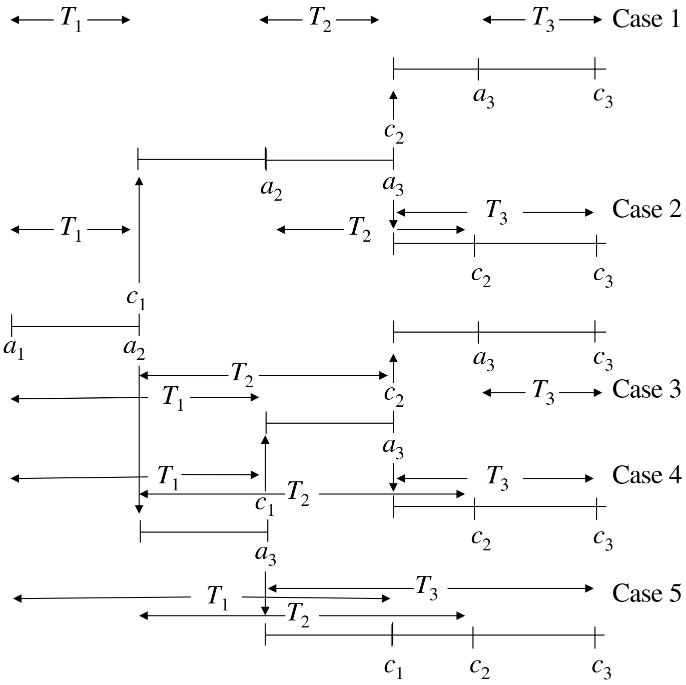

Figure 3: Five cases for $n = 3$ flights' sojourn times in an $M/M/1$ queue.

Using a theorem presented in Kaczynski et al. (2010), the joint PDFs for each of the pairs (T_1, T_2), (T_1, T_3), and (T_2, T_3) in each of the five cases can be determined and then mixed to achieve the three associated joint PDFs. Using these joint densities, the symmetric 3×3 variance–covariance matrix is

$$\Sigma = \begin{bmatrix} \frac{1}{\mu^2} & \frac{\lambda(2\mu+\lambda)}{(\lambda+\mu)^2\mu^2} & \frac{\lambda^2(\lambda^2+4\lambda\mu+5\mu^2)}{(\lambda+\mu)^4\mu^2} \\ \bullet & \frac{2\lambda^2+4\lambda\mu+\mu^2}{(\lambda+\mu)^2\mu^2} & \frac{\lambda(2\lambda^2+8\lambda^2\mu+11\lambda\mu^2+2\mu^3)}{(\lambda+\mu)^4\mu^2} \\ \bullet & \bullet & \frac{3\lambda^6+18\lambda^5\mu+45\lambda^4\mu^2+54\lambda^3\mu^3+30\lambda^2\mu^4+8\lambda\mu^5+\mu^6}{(\lambda+\mu)^6\mu^2} \end{bmatrix}.$$

Substituting $\lambda = 1$ and $\mu = 2$, for example, results in

$$\Sigma = \begin{bmatrix} \frac{1}{4} & \frac{5}{36} & \frac{29}{324} \\ \bullet & \frac{7}{18} & \frac{13}{54} \\ \bullet & \bullet & \frac{1451}{2916} \end{bmatrix} \approx \begin{bmatrix} 0.2500 & 0.1389 & 0.0895 \\ \bullet & 0.3889 & 0.2407 \\ \bullet & \bullet & 0.4976 \end{bmatrix}.$$

These results have been verified via Monte Carlo for the first $n = 3$ flights. The sojourn time variance increases with flight number down the diagonal of the matrix because of the nature of the queueing process, where the sojourn time distribution for each additional flight is dependent on all the previous flights. On the other hand, the off-diagonal covariance entries in each row decrease with flight separation, for example $\sigma_{13} < \sigma_{12}$.

6 EXTENDING FLIGHT COVARIANCE CALCULATIONS

The APPL procedure `Cov(a, b, n)` calculates the covariance between flights a and b ($a < b$) in a system of n flights. For computational considerations (i.e., evaluating the fewest cases necessary for a given n), setting the number of flights $n = b$ provides the fastest result. Additionally, calling `Cov(a, b, n)` where $n > b$ produces a result identical to $n = b$ because flights arriving after flight b do not affect the covariance of previous flights. Using this procedure, the symmetric variance–covariance matrix for $n = 10$ flights with parameters $\lambda = 1$, $\mu = 2$, and $\rho = 1/2$ is calculated and displayed with matrix elements rounded to four significant digits. CPU time is a factor in these computations. Each element in the tenth column of the variance–covariance matrix is calculated from a joint PDF which is a mixture of $C_{10} = 20!/(10!11!) = 16,796$ component distributions, each corresponding to a unique ordering of ATC arrivals and completions.

$$\begin{bmatrix}
0.2500 & 0.1389 & 0.0895 & 0.0621 & 0.0450 & 0.0336 & 0.0257 & 0.0199 & 0.0157 & 0.0125 \\
\bullet & 0.3889 & 0.2407 & 0.1639 & 0.1176 & 0.0872 & 0.0663 & 0.0513 & 0.0402 & 0.0319 \\
\bullet & \bullet & 0.4976 & 0.3251 & 0.2286 & 0.1676 & 0.1263 & 0.0972 & 0.0759 & 0.0600 \\
\bullet & \bullet & \bullet & 0.5845 & 0.3948 & 0.2837 & 0.2113 & 0.1611 & 0.1251 & 0.0984 \\
\bullet & \bullet & \bullet & \bullet & 0.6547 & 0.4524 & 0.3302 & 0.2488 & 0.1915 & 0.1498 \\
\bullet & \bullet & \bullet & \bullet & \bullet & 0.7119 & 0.5000 & 0.3694 & 0.2808 & 0.2177 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0.7587 & 0.5396 & 0.4022 & 0.3080 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0.7974 & 0.5725 & 0.4298 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0.8293 & 0.5999 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0.8559
\end{bmatrix}$$

As the traffic intensity increases, so do the values in the variance–covariance matrix. To illustrate, the same matrix is provided for the increased traffic intensity parameters $\lambda = 1$, $\mu = 10/9$, and $\rho = 9/10$. The increasing sojourn-time variance along the diagonal is expected with the increasing traffic intensity. In addition, the rate that covariance between flights decreases as flight separation increases is less pronounced.

$$\begin{bmatrix}
0.8100 & 0.5856 & 0.4737 & 0.4040 & 0.3553 & 0.3189 & 0.2904 & 0.2673 & 0.2481 & 0.2318 \\
\bullet & 1.3956 & 1.1097 & 0.9393 & 0.8226 & 0.7363 & 0.6692 & 0.6150 & 0.5702 & 0.5323 \\
\bullet & \bullet & 1.9561 & 1.6298 & 1.4167 & 1.2626 & 1.1441 & 1.0494 & 0.9714 & 0.9057 \\
\bullet & \bullet & \bullet & 2.5021 & 2.1458 & 1.8995 & 1.7142 & 1.5679 & 1.4484 & 1.3485 \\
\bullet & \bullet & \bullet & \bullet & 3.0364 & 2.6565 & 2.3831 & 2.1715 & 2.0009 & 1.8593 \\
\bullet & \bullet & \bullet & \bullet & \bullet & 3.5605 & 3.1614 & 2.8652 & 2.6310 & 2.4389 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 4.0754 & 3.6600 & 3.3444 & 3.0904 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 4.5818 & 4.1524 & 3.8199 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 5.0803 & 4.6386 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 5.5713
\end{bmatrix}$$

Traditional steady-state queueing theory and analysis lacks the insight provided in these transient variance–covariance matrices. For military airfields where the number of flights in a day is so small that true steady state is never achieved, routine queueing measures of performance are not representative of reality. Additionally, consider a system where the traffic intensity exceeds one. An appropriate example is the Haiti airport following the 2010 earthquake where military control over the airport was exercised. For a such a system, an analyst might be interested in the covariance between flight sojourn times. Increasing the traffic intensity so that $\rho > 1$ does not preclude covariance calculations using this method, and therefore allows transient analysis of such systems. A variance–covariance matrix for $\rho = 3/2$, is presented below. Given this traffic intensity, the system is unstable and the expected sojourn times for successive flights increase without bound. Along the main diagonal the flight variance is clearly increasing, and the covariance decreases as the separation occurs between flights. This decrease is monotonic, and though not studied in detail here, it appears that the rate of covariance decrease might be of interest for an unstable traffic intensity.

$$\begin{bmatrix}
2.2500 & 1.8900 & 1.7172 & 1.6135 & 1.5438 & 1.4937 & 1.4558 & 1.4263 & 1.4027 & 1.3835 \\
\bullet & 4.1400 & 3.7368 & 3.5018 & 3.3459 & 3.2344 & 3.1507 & 3.0856 & 3.0337 & 2.9913 \\
\bullet & \bullet & 6.0957 & 5.6825 & 5.4166 & 5.2292 & 5.0896 & 4.9817 & 4.8958 & 4.8261 \\
\bullet & \bullet & \bullet & 8.1312 & 7.7208 & 7.4397 & 7.2332 & 7.0747 & 6.9493 & 6.8479 \\
\bullet & \bullet & \bullet & \bullet & 10.2424 & 9.8410 & 9.5538 & 9.3361 & 9.1652 & 9.0276 \\
\bullet & \bullet & \bullet & \bullet & \bullet & 12.4235 & 12.0342 & 11.7463 & 11.5230 & 11.3444 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 14.6687 & 14.2931 & 14.0081 & 13.7828 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 16.9727 & 16.6115 & 16.3319 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 19.3310 & 18.9846 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 21.7397
\end{bmatrix}$$

7 SOJOURN TIME COVARIANCE WITH k FLIGHTS INITIALLY PRESENT

When k flights are present in the $M/M/1$ queue at time zero, the approach used to compute sojourn-time covariance between flights becomes more difficult. When the two flights of interest possess indices larger than k (i.e., T_i where $i > k$), then the approach is similar to that derived in Section 6 of Kaczynski et al. (2010). However, there are two other possibilities. The first possibility is that the first flight has an index of k or less, and the second flight has an index larger than k. In this instance, the only difference in deriving the joint CDF is that the lower indexed flight begins its sojourn time at time zero. In the second possibility, both flights have an index of k or below. If these indices are i and j, where $i < j \leq k$, the time intervals for sojourn times T_i and T_j begin at zero. These calculations are coded in Maple as the procedure `kCov(X, Y, a, b, n, k)`. The first two arguments X and Y are the distribution of time between arrivals, exponential(λ), and the service time distribution, exponential(μ), respectively. The arguments a and b are the flights of interest for the covariance calculation, where $a < b$. The argument n is the number of flights processing through the system not including those present at time zero, which is indicated by the last argument, k. Therefore, the total number of flights processing through the system is $n+k$, and a covariance calculation between any two of these flights can be achieved with the appropriate function call. For example, the function call `kCov(ExponentialRV(1), ExponentialRV(2), 1, 2, 1, 3)` calculates the covariance between flights 1 and 2 in an $M/M/1$ queue with an arrival rate $\lambda = 1$, with service time rate $\mu = 2$, with three flights present at time zero, and a single additional flight processes through the system. The variance–covariance matrix for an $M/M/1$ queue with arrival rate $\lambda = 1$, and service time rate $\mu = 2$, where $k = 4$ flights are present at time zero and an additional $n = 6$ flights process through the system is

$$\begin{bmatrix}
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{211}{972} & \frac{1579}{8748} & \frac{11651}{78732} & \frac{28553}{236196} & \frac{630131}{6377292} & \frac{4646155}{57395628} \\
\bullet & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{211}{486} & \frac{1579}{4374} & \frac{11651}{39366} & \frac{28553}{118098} & \frac{630131}{3188646} & \frac{4646155}{28697814} \\
\bullet & \bullet & \frac{3}{4} & \frac{3}{4} & \frac{211}{324} & \frac{1579}{2916} & \frac{11651}{26244} & \frac{28553}{78732} & \frac{630131}{2125764} & \frac{4646155}{19131876} \\
\bullet & \bullet & \bullet & 1 & \frac{211}{243} & \frac{1579}{2187} & \frac{11651}{19683} & \frac{28553}{59049} & \frac{630131}{1594323} & \frac{4646155}{14348907} \\
\bullet & \bullet & \bullet & \bullet & \frac{37289}{26244} & \frac{271153}{236196} & \frac{1966777}{2125764} & \frac{1588153}{2125764} & \frac{34755203}{57395628} & \frac{763875281}{1549681956} \\
\bullet & \bullet & \bullet & \bullet & \bullet & \frac{1629655}{1062882} & \frac{11663887}{9565938} & \frac{9353743}{9565938} & \frac{203800469}{258280326} & \frac{4465399991}{6973568802} \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \frac{263490131}{172186884} & \frac{208262483}{172186884} & \frac{4506205633}{4649045868} & \frac{98323535707}{125524238436} \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \frac{63939878}{43046721} & \frac{1359189250}{1162261467} & \frac{29402061622}{31381059609} \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \frac{179260456277}{125524238436} & \frac{379721786263}{3389154437772} \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \frac{62708955663745}{45753584909922}
\end{bmatrix}.$$

Here the matrix elements are listed exactly, illustrating Maple/APPL's ability to conduct exact calculations. Unlike the previous variance–covariance matrices, some row elements, in particular those elements associated with flights that are initially present, do not decrease monotonically. These entries are explained in detail in Kaczynski et al. (2010). It can be shown in general that

$$\mathrm{Var}(T_i) = \mathrm{Cov}(T_i, T_j) = \frac{i}{\mu^2}$$

for $i < j \leq k$, where k flights are present at time zero.

8 OTHER POTENTIAL APPLICATIONS

Queueing applications in which the queue behavior does not reach steady-state are commonplace in the military. One potential area of study is the soldier processing center in Kuwait. This center is a clearing house for processing service members in and out of Iraq and Afghanistan. The entire process is complicated, however, individual stations certainly follow the FIFO queue discipline and are good candidates for Markovian queues. This center is known to have substantial problems in (1) arrival rates exceeding service rates, i.e., $\rho > 1$, (2) stations that shut down intermittently without servicing all customers already present, i.e, do not achieve steady state, and (3) form queues prior to servers being available for service, i.e., $k > 0$ customers present at time $t = 0$. Another potential application is installation access through security checkpoints. It is often observed that the arrival rate is substantially higher than the service rate, especially during peak periods. This is better represented as a non-homogeneous Poisson process, and would be an interesting future research topic in transient queueing analysis.

9 CONCLUSIONS AND LIMITATIONS

Previous transient analysis results for the $M/M/1$ and $M/M/s$ queues have been combined with the functionality of the Maple computational engine (and subsequently APPL) in a military air traffic control environment to develop both symbolic and numeric exact flight sojourn time PDFs that can be manipulated to compute and study various measures of performance. A complete variance–covariance matrix for the first $n = 10$ flights and varying traffic intensity is calculated, illustrating this approach's ability to determine the joint PDF between two flight sojourn times. The models presented in this paper have assumed that there are two runways that are dedicated to incoming and outgoing flights which operate independently. Therefore, both arriving flights and departing flights can be viewed separately as independent M/M/1 queues, with the runway playing the role of the server. An analyst would need to assure that data supports the Markovian assumptions, and the policy at the airport supports the FIFO queue discipline. A single-runway airport handling both arriving and departing flights is less likely to conform to the models presented here. A logical epoch for arrival time for an outgoing flight would be the hand off from ground control to tower for outgoing flights; a logical epoch for arrival time for an incoming flight would be initial report to terminal control. Similar conventions would apply when determining the appropriate time for completion of service. The results offer a framework for describing how the well-known $M/M/s$ queue steady-state results occur as the queue progresses toward steady-state. When possible, results are checked against corresponding Monte Carlo simulation and/or previous literature. The first principles derivation suggests a viable alternative for future research would be to apply the approaches provided in this work to a $G/G/1$ queue. Computational considerations take priority as n increases. Making use of other computational formulae (such as Hagwood (2009)) may offer significant time savings and is another interesting avenue for future work.

REFERENCES

Abate, J., and W. Whitt. 1988. Transient behavior of the $M/M/1$ queue via Laplace transforms. *Advances in Applied Probability* 20 (1): 145–178.

e Silva, E., H. Gail, and R. Campos. 1995. Calculating transient distributions of cumulative reward. In *Proceedings of the 1995 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, 231–240. ACM New York, NY, USA.

Gafarian, A., C. Ancker Jr., and T. Morisaku. 1976. The problem of the initial transient in digital computer simulation. *Proceedings of the 76 Bicentennial Conference on Winter Simulation,* 49–51.

Glen, A., D. Evans, and L. Leemis. 2001. APPL: A Probability Programming Language. *The American Statistician* 55 (2): 156–166.

Grassmann, W. 1977. Transient solutions in Markovian queueing systems. *Computers and Operations Research* 4:47–53.

Grassmann, W. 2008. Warm-up periods in simulation can be detrimental. *Probability in the Engineering and Informational Sciences* 22 (3): 415–429.

Hagwood, C. 2009. An application of the residue calculus: The distribution of the sum of nonhomogeneous gamma variates. *The American Statistician* 63 (1): 37–39.

Kaczynski, W., L. Leemis, and J. Drew. 2010. Transient queueing analysis. Technical report, The College of William & Mary, Williamsburg, VA 23187.

Kelton, W. 1985. Transient exponential–Erlang queues and steady-state simulation. *Communications of the ACM* 28 (7): 741–749.

Kelton, W., and A. Law. 1985. The transient behavior of the $M/M/s$ queue, with implications for steady-state simulation. *Operations Research* 33 (2): 378–396.

Law, A. 1975. A comparison of two techniques for determining the accuracy of simulation output. Technical Report 75–11, University of Wisconsin at Madison.

Leguesdron, P., J. Pellaumail, G. Rubino, and B. Sericola. 1993. Transient analysis of the $M/M/1$ queue. *Advances in Applied Probability* 25 (3): 702–713.

Morisaku, T. 1976. *Techniques for data-truncation in digital computer simulation.* Ph. D. thesis, University of Southern California, Los Angeles.

Odoni, A., and E. Roth. 1983. Empirical investigation of the transient behavior of stationary queueing systems. *Operations Research* 31 (3): 432–455.

Parthasarathy, P. R. 1987. A transient solution to an $M/M/1$ queue: A simple approach. *Advances in Applied Probability* 19 (4): 997–998.

Pegden, C., and M. Rosenshine. 1982. Some new results for the $M/M/1$ queue. *Management Science* 28 (7): 821–828.

AUTHOR BIOGRAPHIES

WILLIAM KACZYNSKI is an Assistant Professor in the Department of Mathematical Sciences at the United States Military Academy. He also serves as an officer representative for Army Baseball at the Academy. He holds a B.S. in Economics from the United States Military Academy, a M.S. from the Georgia Institute of Technology in Operations Research, and a Ph.D. from the College of William & Mary in Computational Operations Research. His research interests are in computational probability, random variate generation, and reliability. His email address is <william.kaczynski@usma.edu>.

LAWRENCE LEEMIS is a Professor in the Department of Mathematics at The College of William & Mary. He holds a B.S. in Mathematics from Purdue University, a M.S. from Purdue University in Applied Mathematics, and a Ph.D. from Purdue University in Industrial Engineering. His research interests are in reliability, discrete-event simulation, and computational probability. His email address is <leemis@math.wm.edu>.

JOHN DREW is a Professor Emeritus in the Department of Mathematics at The College of William & Mary. He holds a B.S. in Mathematics from Case Institute of Technology and a Ph.D. in Mathematics from The University of Minnesota. His research interests are in matrix theory and computational probability. His email address is <jhdrew@math.wm.edu>.