Practical Extensions to Cycle Time Approximations for the $G/G/m$-Queue With Applications

James R. Morrison, *Member, IEEE*, and Donald P. Martin

***Abstract*—Approximate closed form expressions for the mean cycle time in a $G/G/m$-queue often serve as practical and intuitive alternatives to more exact but less tractable analyses. However, the $G/G/m$-queue model may not fully address issues that arise in practical manufacturing systems. Such issues include tools with production parallelism, tools that are idle with work in process, travel to the queue, and the tendency of lots to defect from a failed server and return to the queue even after they have entered production. In this paper, we extend popular approximate mean cycle time formulae to address these practical manufacturing issues. Employing automated data extraction algorithms embedded in software, we test the approximations using parameters gleaned from production tool groups in IBM's 200 mm semiconductor wafer fabricator.**

***Note to Practitioners*—We develop extensions to intuitive closed-form approximations for the mean cycle time in queueing networks. Such approximations can be used to analyze the tradeoffs between equipment utilization and cycle time in a manufacturing facility. The extensions incorporate issues of practical import that have not been modeled in the literature and were motivated by the inability of existing models to accurately describe the performance of manufacturing in IBM's 200 mm semiconductor wafer fabricator. The utility of our extensions is that, using automated data collection systems, we are able to well model production tools and elucidate the sources of cycle time.**

***Index Terms*—Production management, queueing analysis, semiconductor device manufacture.**

I. INTRODUCTION

SEMICONDUCTOR wafer fabricators, like many manufacturing systems, suffer from a myriad of complexities not commonly accounted for in traditional queueing models. Explicitly incorporating such complexities into analytic queueing models for system behavior can result in the loss of model tractability. As a consequence, exact or numeric solutions and performance bounds for the cycle time in models such as multiclass queueing networks and $G/G/m$-queues are not commonly employed by practitioners. Rather, simulation and approximate queueing formulae are primarily used to evaluate system cycle time performance. While simulation may allow one to incorporate a diverse array of model features, it seldom provides closed-form expressions for system metrics (such as the mean cycle time), which can be used to understand the behavior. Simple approximation formulae can provide intuition regarding the influence of key parameters on system behavior and, thus, have arisen as a tool of choice at IBM's 200 mm semiconductor wafer fabricator for assessing cycle time performance and identifying improvement opportunities.

Over the years, closed-form rough cut approximations for the mean cycle time of failure prone queues have been developed, see [1]–[6]. For a single-server queue, such approximations agree with exact solutions under specific assumptions on the distributions and failure characteristics ([7]–[9]) and, in general, perform well for multiserver queues. Queueing *networks* have been analyzed using approximate coupling terms intended to capture the essence of interactions between the output of each queue in the network and the variation in the arrivals to other queues [10]–[12].

A practical difficulty with the application of performance evaluation techniques, even when using parameters drawn from actual manufacturing data, is that the manufacturing system's mean performance may not agree with that predicted by the model. One reason for this is that the approximations provide mean cycle time values which deviate from the exact performance of the model. In [12], such errors were frequently less than 10%. Another source of difference, and the problem that we address in this paper, is that there may be *unmodeled* phenomenon which can contribute substantially to fabricator cycle time. In [13]–[20], the authors attempt to bridge the gap between measures of actual system performance and mean values predicted by simple queueing models such as the M/D/1 queue or, more generally, the $G/G/m$-queue. This paper incorporates four practical manufacturing realities into a common closed-form approximation for the mean cycle time behavior of a $G/G/m$-queue in an attempt to expand the practical applicability of existing approximations. Some of the results of this paper first appeared in [21].

In Section II, we recall approximate cycle time formulae and bounds for $G/G/m$-queues subject to server failure. Our intent in this section is to provide a brief survey of exact results, popular approximations and bounds possessing simple closed-form expressions, and demonstrate the efficacy and intuitive nature of the approximations. We extend the approximations for tools possessing parallel processing capabilities in Section III. This extension is essential since many important classes of tools in semiconductor manufacturing possess parallel processing capabilities (e.g., photolithography cluster tools, copper plating tools, and ion implant tools). Section IV incorporates the time

Manuscript received July 21, 2006; revised February 2, 2007. This paper was recommended for publication by Associate Editor J. G. Shanthikumar and Editor N. Viswanadham upon evaluation of the reviewers' comments.

J. R. Morrison is with the Department of Engineering and Technology, Central Michigan University, Mount Pleasant, MI 48859 USA (e-mail: morri1j@cmich.edu).

D. P. Martin is with the Systems and Technology Group, IBM Corporation, Essex Junction, VT 05452 USA (e-mail: martindp@us.ibm.com).

Digital Object Identifier 10.1109/TASE.2007.905975

1545-5955/$25.00 © 2007 IEEE

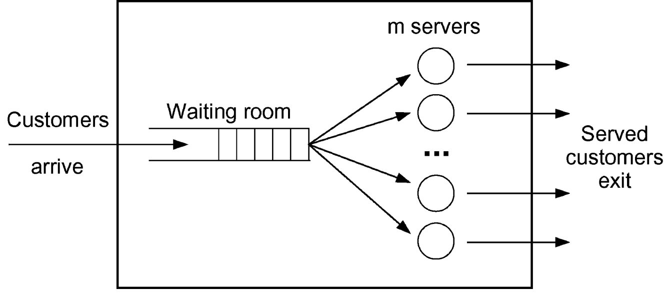

Fig. 1. A multiserver queue.

that a tool remains idle even when available work is present (often referred to as idle with work in process). Cycle time offsets, such as travel time from one queue to another or hold time (during which a lot is removed from the production queue, possibly pending the resolution of a process issue), are addressed in Section V. In Section VI, we incorporate tool failure behavior allowing for a lot to defect from a failed server and return to the queue (it can be inferred that typical cycle time approximations assume that lots are loyal to a failed server) as may be common for certain tools or lengthy down time events. The implementation of the approximations to tool sets from IBM's 200 mm semiconductor wafer fabricator is discussed in Section VII. There we demonstrate that the new formulae can well serve to elucidate the sources of cycle time in a practical manufacturing system. Concluding remarks are presented in Section VIII.

II. Approximations and Bounds for the Mean Cycle Time

Consider a system in which customers arrive to a waiting room (or queue) of infinite size that is catered to by m identical servers or tools. The service time of a customer at a tool is a random variable with general distribution and mean $1/\mu$ (every service time is drawn from the same distribution). The interarrival times between customer arrivals to the queue are random variables with general distribution and mean $1/\lambda$ (every interarrival time is drawn from the same distribution which may differ from the service distribution). All service and interarrival times are assumed independent. Arriving customers are served in a first-come first-served manner (FSCS), though this restriction can be relaxed (so long the process time for a customer is not part of the decision). An idle tool accepts a waiting customer as soon as one is available, only one customer may receive service from a tool at a given time, and only one tool may attend to each customer at a time. Such a system is often ([6], [7]) referred to as a $G/G/m$-queue, following Kendall's notation [22] (the first and second G indicate the generally distributed interarrival and service times, respectively, and m is the number of servers). Fig. 1 depicts such a queue. Hereafter, we refer to customers as lots (a lot in semiconductor wafer manufacturing refers to a container filled with up to 25 wafers) and servers as tools.

A. *Cycle Time Approximations and Bounds*

An important measure of performance in a queueing system is the total time that a lot spends in the system, termed the cycle time. For a $G/G/m$ queue, cycle time consists of queue time and service time. Based on the work of [1], [2], and [11], the following approximation for the mean cycle time $E(CT)$ in a $G/G/m$-queue has been proposed (see [6]):

$$E(CT) \approx \frac{1}{\mu} + \frac{1}{\mu}\left(\frac{c_A^2 + c_S^2}{2}\right)\left(\frac{\rho^{-1+\sqrt{2m+2}}}{m(1-\rho)}\right). \quad (1)$$

Here, the system loading $\rho = \lambda/(m\mu)$ is assumed less than one, m is the number of servers, c_A is the coefficient of variation of the interarrival time ($c_A = \sigma_A/(1/\lambda)$, where σ_A is the standard deviation of the interarrival time), and c_S is the coefficient of variation of the service time ($c_S = \sigma_S/(1/\mu)$, where σ_S is the standard deviation of the service time). Note that as the number of servers m increases, the effect of loading on the queue time is reduced. Also, as the coefficients of variation for the interarrival and service times increase, so too does the approximation for the queueing. The intuitive value of such a formula is clear. Note that the bounds of [4] could be employed instead of (1) to possibly obtain a tighter approximation.

It is interesting to compare the mean cycle time prediction of this simple approximation against exact performance and simple closed-form bounds. Based on the work of [23] and [24], queueing texts [25] provide performance bounds for $G/G/m$-queues

$$E(CT) \le \frac{1}{\mu} + \left[\frac{\rho}{1-\rho}\right]\left[\frac{c_S^2\left(2-\frac{1}{m}\right) + \frac{c_A^2}{(m\rho^2)} + \frac{(m-1)}{m}}{2\mu}\right].$$

A lower bound may be computed [23]–[25] as

$$CT_1 - \frac{(m-1)\mu E(S^2)}{2m} \le E(CT)$$

where S is the random variable for the service distribution, $E(S^2)$ is its second moment, and CT_1 is the expected cycle time for a $G/G/1$-queue with arrival process as in the $G/G/m$-queue and service process given by the random variable $S^* = S/m$ (so that $E(S^*) = 1/(m\mu)$ and $\sigma_{S^*} = \sigma_S/m$). In the event that an exact expression is not known for CT_1, a lower bound can be used. When the lower bound is less than the service time, one can use the service time instead.

The upper and lower performance bounds along with two approximations for the mean cycle time of an $M/M/2$-queue are shown in Fig. 2 as a function of the loading ρ. Since there is a readily calculable explicit solution for the mean cycle time [7], [25], one is able to compare the performance of the quartet of the bounds and approximations to the true behavior. The Martin approximation is the name we at IBM Vermont have given to the formula

$$E(CT) \approx \frac{1}{\mu} + \frac{1}{\mu}\left(\frac{c_A^2 + c_S^2}{2}\right)\left(\frac{\rho^m}{(1-\rho^m)}\right)$$

which is *exact* for the $M/M/2$-queue. This expression, which is simpler and less accurate, but perhaps more intuitive, than the approximations of (1) or [5], suggests that the mean queueing time in a multiserver queue is approximately that of a single-server queue with loading ρ^m. Note that the bounds depicted in Fig. 2 only become meaningful in heavy traffic (as $\rho \to 1$). The approximations of [6] are within 3% of the exact value and,

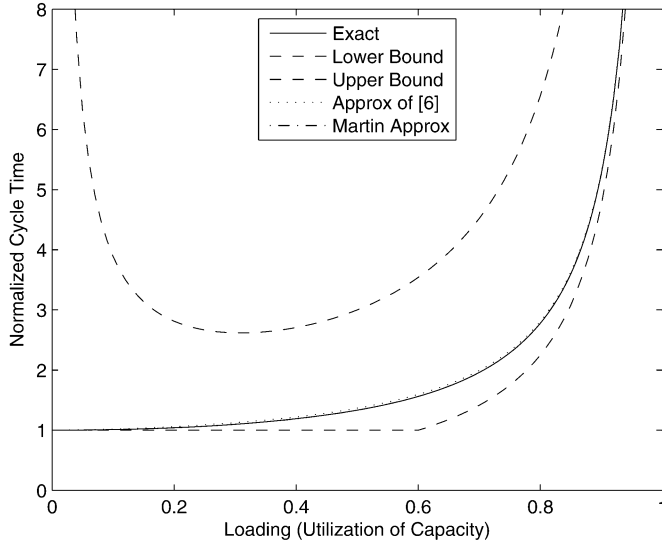

Fig. 2. Bounds, approximations, and the exact solution for the cycle time of an $M/M/2$-queue.

together with the Martin approximation, are virtually indistinguishable from the exact solution at the (very reasonable) scale employed.

It is important to note that there are other approximations for the mean cycle time in $G/G/m$-queues that can be more accurate, such as those of [4] and [5]. There is, however, an increase in the computation required to evaluate such approximations and the expressions can be less intuitive.

B. *Tool Failure With Loyal Lots*

Queues with failure prone tools have been studied since [26]. For the $M/G/1$-queue with its tool subject to independent exponentially distributed available time and generally distributed repair intervals, solutions for system performance have been obtained; see, for example, [8], [27], and [28]. For generally distributed times until failure, bounds have been derived, see [29]. For a queue with a single tool, the literature addresses different assumptions for how lots respond to a tool failure. Two of the common behaviors are preempt-resume and preempt-repeat. The preempt-resume behavior assumes that the service time devoted to a lot whose service is interrupted (preempted by tool failure) is not lost and service is resumed once the tool returns from failure. Preempt-repeat behavior assumes that service time is lost and the production run must be started from the beginning when the tool returns.

Consider such an $M/G/1$-queue, subject to preempt-resume failures, with exponentially distributed available intervals of mean m_F and generally distributed failure intervals with mean m_R and coefficient of variation c_R. The mean availability of the server A is, thus, $A = m_F/(m_F + m_R)$. Let λ be the arrival rate of lots, μ be the service rate and c_S be the coefficient of variation of the service times as before. The mean cycle time (see [8]) is

$$E(CT) = \frac{1}{\mu A} + \frac{1}{\mu A}\left(\frac{\rho^*}{1-\rho^*}\right)\left(\frac{1}{2}\right) \times \left(1 + c_S^2 + \frac{\left(1+c_R^2\right)A(1-A)m_R\mu}{\rho^*}\right)$$

where $\rho^* = \lambda/(\mu A)$ satisfies $0 \leq \rho^* \leq 1$. Note that as the loading approaches 0, one expects no queueing and the mean cycle time has the form

$$\lim_{\rho\to 0_+} E(CT) = \frac{1}{\mu A} + (1-A)\frac{m_R\left(1+c_R^2\right)}{2} \tag{2}$$

where 0_+ indicates the right-sided limit.

The expression of (2) for the mean cycle time in the low loading regime is not surprising on account of the following conceptual argument (which elucidates how the terms arise and will guide our development later). Consider a lot arriving randomly to an empty queue (zero loading implies zero probability of queueing). With probability $1 - A$, the arriving lot faces a failed tool and must wait a certain amount of time for that tool to return to service [let E (residual repair time) denote its mean]. Once the tool is available, the lot requires on average $1/\mu$ time units of service from the tool. During this time, the tool may fail a number of times, and each such failure adds an additional repair time (with mean m_R) to the overall cycle time. The number of failures during the process time is the number of counts of a Poisson process of rate $1/m_F$ in the time interval $1/\mu$ (since the time until a failure is exponentially distributed with rate $1/m_F$). Combining these terms gives

$$\lim_{\rho\to 0_+} E(CT) = (1-A)E[\text{residual repair time}] + \frac{1}{\mu} + m_R\sum_{k=0}^{\infty} k\text{Prob}(k \text{ failures in } 1/\mu).$$

The summation is just the *expected* number of counts of the Poisson process, which is $(1/m_F)(1/\mu)$. Hence

$$\lim_{\rho\to 0_+} E(CT) = \frac{1}{\mu A} + (1-A)E[\text{residual repair time}].$$

The mean residual repair time is $(m_R/2)(1 + c_R^2)$, see [7], so that this expression agrees with (2). Intuitively, the low loading cycle time consists of the expected time until the server is first available plus the amount of time it takes the server to complete the work given that on average the server is available a proportion of time equal to A.

Approximations have been proposed for $G/G/1$-queues with preempt-resume tool failure (exponential time to failure and general repair distribution). The following is suggested in [6]:

$$E(CT) \approx \frac{1}{\mu^*} + \frac{1}{\mu^*}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{\rho^*}{1-\rho^*} \tag{3}$$

where $\mu^* = \mu A$ and the *effective* coefficient of variation of the service time $c_{S,E}^2$ (intended to capture the essence of the system behavior in response to less than perfect server availability) is

$$c_{S,E}^2 := c_S^2 + \left(1+c_R^2\right)A(1-A)m_R\mu.$$

Fig. 3 compares the exact performance of an $M/M/1$-queue subject to tool failures with the approximation of (3). Here the failure and repair times are exponentially distributed, preempt-resume service is employed, $m_F = 16$, $m_R = 4$, and $\mu = 1$. The squared coefficients of variation are all 1.0 for the exponential distributions and the average availability of the tool is

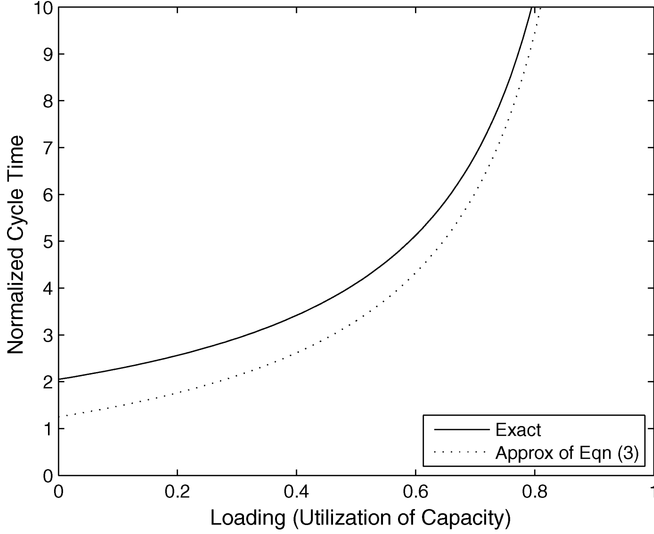

Fig. 3. Exact performance and an approximation for the cycle time of a failure prone $M/M/1$-queue.

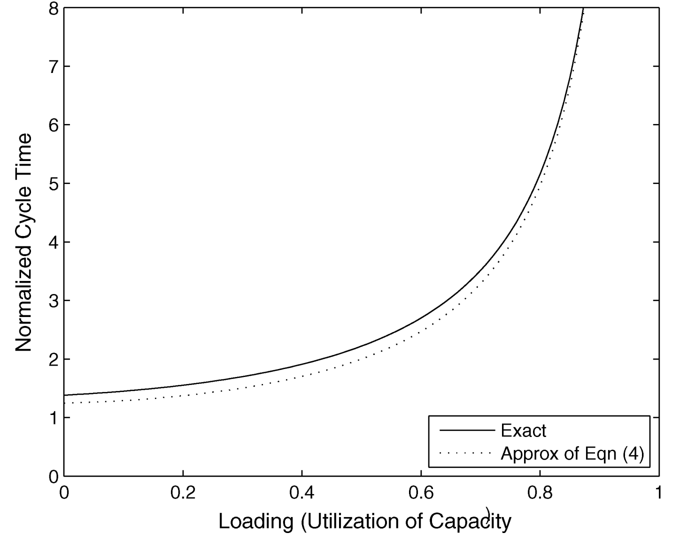

Fig. 4. Exact performance and approximations for the cycle time of a failure prone $M/M/2$-queue.

$A = m_F/(m_F + m_R) = 80\%$. Particularly in the medium to low loading regime, the approximation is unsatisfactory. Note that it is in this regime that one is cautioned about using such approximations [4].

For multiple tools serving the queue, assuming preempt-resume tool failure behavior, the approximation of [6] becomes

$$E(CT) \approx \frac{1}{\mu^*} + \frac{1}{\mu^*}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{(\rho^*)^{-1+\sqrt{2m+2}}}{m(1-\rho^*)} \tag{4}$$

where $\mu^* = \mu A$ and $\rho^* = \lambda/(m\mu^*)$. As the loading $\rho \to 0^+$, $E(CT) \to 1/(\mu A)$. From this we infer that the approximation should serve best when the lots are assumed loyal to a failed server, though the residual repair time is not included. One might also consider this approximation as a compromise between the behavior of loyal lots and lots that defect from a failed server and return to the queue (Section VI discusses the distinction in more detail).

Fig. 4 compares the exact [9] and approximate expressions above for the mean cycle time performance of an $M/M/2$-queue subject to tool failures. The failure and repair times are exponentially distributed, preempt-resume service is employed, lots defect from a failed server and return to the head of the queue, $m_F = 16$, $m_R = 4$, and $\mu = 1$. The squared coefficients of variation are all 1.0 for the exponential distributions and the average availability of the tool is $A = m_F/(m_F + m_R) = 80\%$. The approximations perform reasonably well (see Section VI).

III. Tools with Parallelism

Many tools employed in the manufacture of semiconductor wafers conduct multiple operations on each wafer sequentially within the same chassis. A lot may enter production once all wafers of the preceding lot have completed the first operation. A special class of such tools is the cluster tool which can serve to model the photolithography cluster tool [16], [18]. For brevity, let all lots consist of L wafers. Consider a tool comprised of M modules, each of which can provide a distinct necessary operation for up to W wafers simultaneously (*require that L mod $W = 0$, preventing partially filled module*). Let Δ denote the deterministic process time for the W wafers in a module. A lot is processed in groups of W wafers, each of which proceeds sequentially through the modules in the tool as they are available. When the first module completes the last W wafers in a lot and is vacated, the subsequent lot may enter production. Refer to such a tool as a *parallel processing* tool.

Label lots in the order in which they are processed, so that lot ℓ_{i+1} follows lot ℓ_i. Letting μ denote the maximum throughput rate (*lots* per unit time), m_S denote the total service time for a lot, c_i denote the completion time of lot ℓ_i, and b_i denote the time lot ℓ_i begins production, we have

$$\begin{aligned}
\mu &= \frac{W}{L\Delta}\\
m_S &= \left[\left(M + \frac{L}{W}\right) - 1\right]\Delta\\
T_i &:= \mathrm{Min}\{m_s, c_i - c_{i-1}\} \in [(L/W)\Delta, m_s]\,, \text{and}\\
\phi_i &:= \frac{m_S}{T_i} \in \left[1, 1 + (M-1)\frac{W}{L}\right].
\end{aligned}$$

T_i is the time between the completion of lots, so long as the tool is not idle between them. The symbol ϕ_i is, thus, a measure of the achieved production parallelism. The parallelism achieves its maximum value when a lot is started immediately upon vacancy of the first module. Use ϕ to denote the maximum parallelism, so $\phi = 1 + (M-1)(W/L)$. Note that $1/\mu$ is not equal to m_S.

The following theorem is suggested in [16] and [17] and proved in [18]. It is an extension of [30] to batch arrivals.

Theorem 1: The mean cycle time for a parallel processing tool with exponential interarrival times of rate λ and *deterministic process times* is

$$E(CT) = \frac{\phi}{\mu} + \left(\frac{1}{\mu}\right)\left(\frac{1+c_S^2}{2}\right)\left(\frac{\rho}{1-\rho}\right). \tag{5}$$

Here, $\rho := \lambda/\mu$ and $c_S^2 = 0$ for the deterministic process time.

We explicitly include the c_S^2 term (even though its value is 0 in the theorem) to suggest the appropriate place for such a term for the case of more general service distribution. Note that the queueing term is proportional to the time that it takes for a lot to vacate the first module $(L/W)\Delta$ (thereby allowing another lot to enter production). In fact, this time is the inverse of the maximum production rate μ.

The insight of Theorem 1 is that, for deterministic module process times, the first module behaves as an $M/D/1$-queue with service rate μ. The subsequent modules simply add additional service time, but are irrelevant to the queue.

As a first-order attempt to incorporate parallelism into our approximations for a failure prone $G/G/m$-queue served by m parallel processing tools (each with mean service time m_S taken from the general distribution and maximum parallelism ϕ, so that $\mu = \phi/m_S$) with arrival rate λ, we resort to the following two assumptions.

Assumption A1: The time for lot ℓ_i, with service time X_i, to exit the first module is given as X_i/ϕ.

Assumption A2: Each lot, upon exit from the first module, is considered to depart the system. The parallelism is modeled as additional service time (equal to $X_i - X_i/\phi$) subsequent to the departure and independent of the system state. This additional time is delayed each time the server fails.

Queueing time is unchanged and (4) becomes

$$E(CT) \approx \frac{\phi}{\mu A} + \frac{1}{\mu A}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{(\rho^*)^{-1+\sqrt{2m+2}}}{m(1-\rho^*)} \quad (6)$$

where $\rho^* = \lambda/(m\mu A)$ and $c_{S,E}^2$ is as in Section II. The use of the same $c_{S,E}^2$ follows since, if m_S and σ_S denote the mean and standard deviation of the process time, respectively, then the mean and standard deviation of the time to vacate the first module is m_s/ϕ and σ_S/ϕ. The coefficient of variation of the time to vacate the first module is $(\sigma_S/\phi)/(m_s/\phi) = c_S$.

We thus model the cycle time by considering the first module to be a $G/G/m$-queue and incorporate the parallelism as an added post production increase in the process time (which is delayed when its server fails). Of course, this blithely ignores the fact that for true parallel processing tools the modules after the first are shared by all lots on the tool and the interactions can not truly be ignored. Thus, variation in process times and nonideal tool availability will wreak havoc on the neat formula (5). However, the approximation is exact for $m = 1$ and $A = 1$, if the service time is deterministic and interarrival times exponential, and does begin to capture important features of parallel processing tools in the general context.

IV. Idle With Work Available

In many manufacturing systems, including both manual and automated systems, tools may lie fallow while awaiting lots that are considered to be in queue (by virtue of having completed the previous stage of production). Such a tool is said to be *idle with work in process*, or idle with WIP. An idle with WIP condition can occur for many reasons, including lack of an operator and delays between the arrival of a lot to the tool area and its subsequent loading onto the tool (e.g., the automated material handling system may only begin to move a lot to a tool once the previous lot has vacated the tool). Algorithms for measuring the duration of idle with WIP have been proposed in [13]. There the idle with WIP time is termed operator and deployment loss.

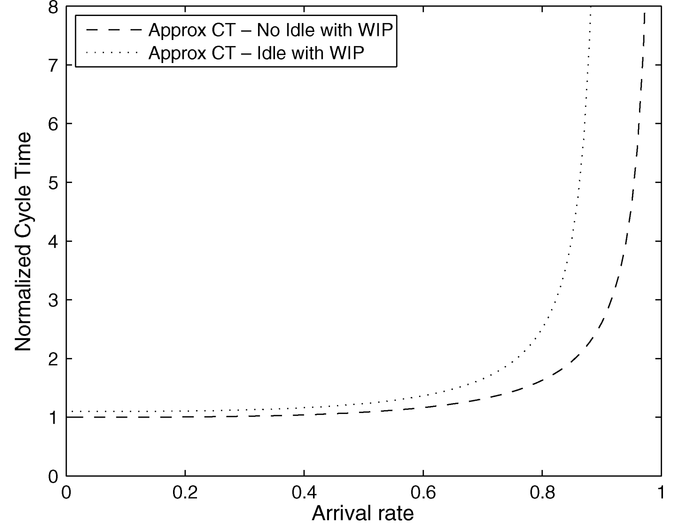

Fig. 5. Approximate cycle time performance for a $G/G/m$-queue with and without idle with WIP.

Suppose that each lot experiences a random preproduction delay which is independent of the production time itself, during which time the tool is idle. Let m_I and σ_I denote the average and standard deviation, respectively, of the idle with WIP time associated with the loading of each lot. To incorporate this feature into a $G/G/m$-queue with parallel processing tools, we consider the idle with WIP to be an independent random increase in the process time (note that this is a special case of the nonpreemptive outages discussion of [6, p. 258]). As a consequence, the *effective* process rate μ_e and effective squared coefficient of variation become

$$\mu_e := \left[\frac{1}{\mu A} + \frac{m_I}{A}\right]^{-1}, \text{and}$$

$$c_{S,E}^2 := \frac{(\sigma_S/\phi)^2 + \sigma_I^2}{\left[(1/\mu) + m_I\right]^2} + \left(1 + c_R^2\right)A(1-A)\frac{m_R}{(1/\mu) + m_I}$$

where σ_S is the standard deviation of the service time. The resulting cycle time approximation for a $G/G/m$-queue with idle with WIP is

$$E(CT) \approx \frac{(\phi/\mu) + m_I}{A} + \frac{1}{\mu_e}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{(\rho_e)^{-1+\sqrt{2m+2}}}{m(1-\rho_e)} \quad (7)$$

where $\rho_e = \lambda(m_I + 1/\mu)/(mA)$ and all variables not defined in this section are as in Section II. Note that the effective process time $1/\mu_e$, and as a consequence the effective loading ρ_e, are increased above the case when there is no idle with WIP (i.e, $m_I = 0$). The resulting cycle time curve is thus shifted "up and to the left," as depicted in Fig. 5. There, perfect tool availability is assumed ($A = 100\%$), $\mu = 1$, $\phi = 1$, $m = 3$, $m_I = 0.1$, $c_{S,E}^2 = 0.2$, and $c_A^2 = 1$. The effective process time is thus 1.1 time units per lot, while $1/\mu = 1$ time unit per lot.

While the time that tools are idle with WIP in queue can be measured [13], it generally consists of three parts. The first, which is modeled above, is caused by activities that cannot be avoided or conducted while the tool is busy. Examples include time required to load a tool with a single load port (when the load time is not considered as part of the production time) or a setup that must be conducted after the lot is loaded on the tool. The second source of measured idle with WIP time is caused by activities that can be performed while the tool is busy, but are not. These include administrative tasks that an operator may be required to perform as a routine part of the job. A third source of idle with WIP time is insufficient track or operator resources resulting in contention that leaves the tool idle.

Typically, as the loading and cycle time on a toolset increase, the operator or track resources dedicated to serving the toolset also increase. In addition, tasks that can be performed while the tool is busy are conducted more often during tool busy time at increased loading. As a consequence, the measured idle with WIP time decreases at higher loading! Thus, a measured value of idle with WIP obtained at a specific loading ρ_e will not necessarily apply for a different value of loading. One could consider this as rendering the value of m_I a function of loading. In general, we will ignore this fact and assume the m_I is constant for purposes of generating performance curves which are a function of loading.

V. Cycle Time Offsets

In many manufacturing facilities, production involves activities not requiring the capacity of a wafer processing tool. Such activities may include the travel of lots from one tool group to the next, removal of a lot from the queue pending the resolution of a process issue (termed *hold*), or delay in removing the lot from the tool once production has ended. These activities can often be considered as independent of the queue length of the tool group. As such, the total mean cycle time for a lot requiring service from a tool group is the mean cycle time incurred during travel, hold and post production unload activities plus the mean cycle time incurred in queueing and process time.

Let T denote the mean time spent by a lot traveling from the previous stage of production to the one of interest, including queue time for the travel service and travel time but *excluding* possible time waiting for the destination tool group to request the lot (this measurement perspective is key to allowing one to suppose independence of the travel time from the tool queue). Let H denote the mean time spent by a lot on hold. Denote by P the mean time each lot waits to be removed from the tool (and is thereby unavailable to begin transport to the subsequent stage of production). Assuming independence of the cycle times at each phase, the approximation of (7) becomes

$$E(CT) \approx T + H + P + \frac{(\phi/\mu) + m_I}{A} + \frac{1}{\mu_e}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{(\rho_e)^{-1+\sqrt{2m+2}}}{m(1-\rho_e)} \quad (8)$$

where all variables have been defined previously. The approximation suggests that if the independent cycle time offsets are reduced, the overall cycle time will improve. The consequence of this for a factory, which consists of many such tools, is that if the average travel time or hold time in the entire factory can be reduced the overall cycle time of the entire facility should improve.

VI. Defection of Lots From a Failed Tool

As discussed in Section II, the previous approximations may perform poorly in the low to moderate loading regime when the tools are failure prone. Note that when the loading $\rho_e \to 0+$ (approaches 0 from the right) and ignoring cycle time offsets and idle with WIP, (4) provides the cycle time approximation of $(1/\mu)/A$. If the tool average availability is 80% (not an unreasonable number for an implant tool including planned and unplanned down time), the cycle time prediction is thus $1.25/\mu$. This prediction is independent of the number of tools, a general behavior which is correct when lots are loyal to a failed tool. However, when lots are allowed to *defect from a failed server and return to the head of the queue* (where they may immediately continue service with another server if one is available), the low loading cycle time should be a function of the number of servers.

Allowing lots to *defect from a failed server and return to the head of the queue*, for low loading (there is small probability of queueing behind other lots) an arriving lot expects that there is only a delay in entering production when *all* servers fail. Once production begins, the average proportion of time that at least one server is available is $(1 - \text{Prob[all are down]}) = 1 - (1 - A)^m$. Thus, we roughly expect that

$$\lim_{\rho \to 0^+} E(CT) \approx \left(\frac{1/\mu}{1-(1-A)^m}\right) + (1-A)^m E[R_m]$$

where R_m is the residual time until at least one server returns from failure, given that all are down.

Final Approximation: The mean cycle time for a $G/G/m$-queue with exponential time to tool failures, general repair times, preempt-resume processing, lot defection from failed servers to the head of the queue, parallel processing tools with maximum parallelism ϕ, idle with WIP, and cycle time offsets is approximately

$$E(CT) \approx (T + H + P) + (1-A)^m E(R_m) + \frac{(\phi/\mu) + m_I}{1-(1-A)^m} + \frac{1}{\mu_e}\left(\frac{c_{S,E}^2 + c_A^2}{2}\right)\frac{(\rho_e)^{-1+\sqrt{2m+2}}}{m(1-\rho_e)}. \quad (9)$$

Here, the effective process rate and effective squared coefficient of variation are as in Section IV.

Note that the idle with WIP portion of the effective process rate has also been deflated here. This agrees with the interpretation of the idle with WIP as a *one time* preprocessing. If the idle with WIP is, in fact, to be interpreted as loading the tool, our cycle time approximation may be too low. Since each time the server fails, the idle with WIP will occur again when loading the lot on another tool. The assumption regarding the character of the idle with WIP will not matter when we implement the approximations as the idle with WIP time associated with each lot will be explicitly measured (it will not then matter how it arises).

TABLE I
MEAN OF MINIMUM OF RESIDUAL REPAIR TIMES—$E[R_m]$

Repair Time Distribution	$E[R_m]$
Deterministic	$\frac{m_R}{m+1}$
Exponential	$\frac{m_R}{m}$

The residual time until a server is available $E(R_m) \approx E[\min(X_1, \ldots, X_m)]$, where X_i is the residual repair time of the ith tool. Let $F_R(t)$ denote the cumulative distribution function of the repair time. The residual repair time X_i has probability density function $\hat{f}_R(t)$ given as (see [7, p. 172])

$$\hat{f}_R(t) = \frac{1 - F_R(t)}{m_R}.$$

In general, $E(R_m)$ may not be easy to calculate. Table I provides values for the cases of deterministic ($c_R = 0$) and exponential ($c_R = 1$) repair times. For practical purposes, one could use these values to approximate the value for an unknown repair time distribution with calculable c_R.

The approximation of (9) incorporates many features of practical import which have not, to the authors' knowledge, been included in existing models. In the subsequent section, the efficacy of the new approximations are demonstrated by application.

VII. Implementation of the Approximations

The extensions to cycle time approximations proposed were inspired by the study of tool groups in IBM's 200 mm wafer fabrication facility. Numerous studies had been conducted and algorithms developed to close the gap between observed tool performance and predictions given by the $M/D/1$-queue model, or more generally, the $G/G/m$-queue model (see, for example, [13]–[15] and [18]–[20]). Yet, for many tool groups, the cycle time approximation formulae did not adequately reflect the measured system behavior. A primary reason for the remaining discrepancies was unmodeled system dynamics. We have attempted to capture the essence of these dynamics with our extensions.

A. Assumptions and Measurement of Parameters

There are three notable caveats and some further comments on parameter estimation worth mention prior to delving into implementation examples.

First, no tool group at IBM's 200 mm wafer fabricator (even with the extensions) is actually a $G/G/m$-queue operating in equilibrium. Lot process times are not drawn from a single probability distribution, but rather are close to deterministic with duration dependent upon the stage of production. The tool groups are not independent of each other; each is part of a queueing network. Further, fabricator demand cycles subject each group of similar tools to time varying arrival rates.

Second, it can be difficult to determine the number of servers that may provide service to a given lot. For example, even though a fleet of ion implant tools may contain say a dozen similar tools, each lot may only be officially qualified (for yield loss detection) to run on three of the fleet at a given stage of production. Further, tools may not be grouped in the same geographic area and even if they are, bays and operator assignment can create virtual subsets of tools. As a consequence, a small number of servers are typically available to a given lot (a fleet of similar tools operates as a collection of queues, each with a much reduced number of servers).

Third, though the approximate cycle time curves are a function of loading, we have assumed that no parameter changes with the loading. One parameter which we have observed to change with loading is idle with WIP, as discussed in Section IV. The performance approximation remains accurate for a given loading level studied, however, one must remain cognizant that parameters could change with loading. Note that the coupling terms of [11] and [12] use a loading dependent coefficient of variation of the interarrival times.

In IBM's 200 mm wafer fabricator, the CACTUS measurement system [13] enables the acquisition of the required statistics and parameters. The measurement algorithms determine the idle with WIP experienced by each lot during its sojourn through a given tool group. As such, there is no need to specify the character of the idle with WIP, as discussed in Section VI. In addition, means of the process time (ϕ/μ), travel time (T), hold time (H), post production unload time (P), loading (ρ_e) (mostly), and availability (A) are calculated automatically by the data acquisition systems. Other data was calculated manually.

The parameters used in the examples of Sections VII-B and VII-C were extracted from a two week time period. The parameters were then used to generate a performance approximation as a function of loading (9) and the achieved mean cycle time *during the same period* was measured. Longer time frames enable one to more accurately capture random events which occur at slow time scales. However, many of the random events occur with mean on the order of one hour. Hence, observing a two week period for multiple tools enables us to observe on the order of thousands of events. The parameters used to construct the mean cycle time approximation in the example of Section VII-D were obtained from about one year of data; the actual performance points were observed during that time period. In the first example of Section VII-B, we show the calculations necessary to reach the final mean cycle time approximation. Otherwise, we immediately report the final approximation.

B. Copper Plating Example

The normalized parameters provided in Table II were obtained from the study of two weeks of data from a copper plating tool group in IBM's 200 mm wafer fabricator. We purposely omit the manufacturer of the tool and the time frame under consideration. There are six tools in the fleet, roughly segregated into two (not too distant) groups.

With these parameters, one obtains

$$\frac{1}{\mu_e} = \left[\frac{1/\mu + m_I}{A}\right] = \left[\frac{0.833 + 0.287}{0.933}\right] = 1.201$$

$$c_{S,E}^2 = \frac{(\sigma_S/\phi)^2 + \sigma_I^2}{[(1/\mu) + m_I]^2} + \frac{\left(1 + c_R^2\right) A(1-A)m_R}{(1/\mu) + m_I} = 0.701.$$

TABLE II
MEASURED TOOL PARAMETERS

Parameter	Copper Plating
T	0.830
H	0
P	0.226
c_A^2	1.16
$1/\mu$	0.833
ϕ/μ	1
σ_S^2	0.127
ϕ	1.20
m_I	0.287
σ_I^2	0.077
A	0.933
m_R	4.02
c_R^2	1.54
m	6

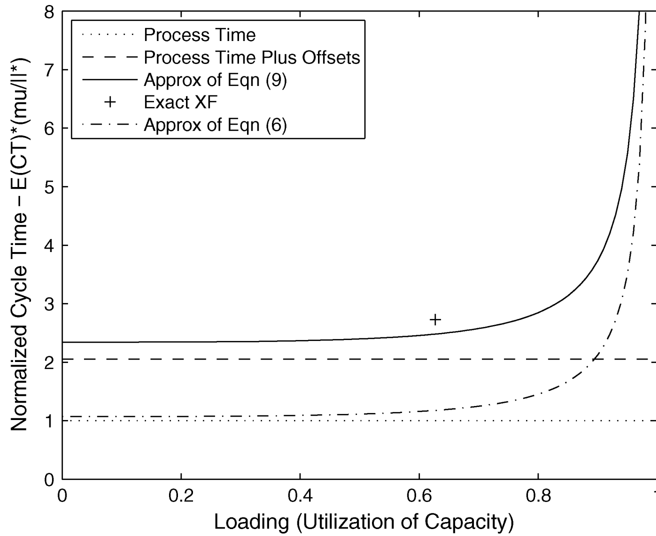

Fig. 6. Normalized cycle time (XF) prediction and actual performance for a copper plating toolset with six servers. The approximation is within about 9% of actual.

Setting the value of $E[R_m] = m_R/m = 0.670$ time units, the cycle time approximation of (9) becomes

$$E(CT) \approx (1.055) + (0) + (1.287) + (1.201) \left(\frac{1.86}{2}\right) \frac{\rho_e^{2.74}}{6(1-\rho_e)}.$$

For the *same time period* from which the parameters were obtained, the measured loading was $\rho_e = 0.627$ and the measured normalized mean cycle time was 2.73 (when normalized by the mean process time, cycle time is termed flow factor or X-factor). The actual performance and the approximate performance curves are depicted in Fig. 6. There, the first horizontal line indicates the mean process time and the second the mean process time plus the mean cycle time offsets. The curve above the lines indicates the additional cycle time incurred due to idle with WIP and queueing. For comparison, the approximation of (6), which includes only the parallelism feature, is plotted as well. Note the difference between the approximations.

TABLE III
NORMALIZED CYCLE TIME FOR COPPER PLATING

Model	$E(CT) * (\mu/\phi)$	|% Error |
$M/D/1$-queue	1.84	32.6%
Approx of Eqn (6) with m = 6	1.18	56.8%
Approx of Eqn (6) with m = 3	1.40	48.7%
(9) with m = 6	2.48	9.2%
(9) with m = 3	2.77	1.4%
Actual	2.73	-

Table III compares the normalized mean cycle time predictions for an $M/D/1$-queue, the failure prone $G/G/m$-queue approximation of (6) (which includes only the parallelism feature) and the extended approximation of (9) with the actual performance for the loading $\rho_e = 0.627$. All models include the idle with WIP in the effective process rate. The prediction of (9) performs well. Note that the number of servers appears too high, perhaps on account of the fact that the tool group has some geographical separation.

C. Many Server Chemical Mechanical Polish (CMP) Example at High Loading

In this section, we study a chemical mechanical polish (CMP) tool group in IBM's 200 mm wafer fabricator consisting of 38 *essentially identical tools* grouped into a single geographical area. The performance parameters used were obtained from two weeks of data. We purposely omit the type of polish, the manufacturer of the tool, and the time frame under consideration. In addition, the time units have been normalized.

The cycle time approximation of (9) is

$$E(CT) \approx (2.58) + (3.69) \frac{\rho_e^{7.83}}{38(1-\rho_e)}.$$

The measured loading for the time period from which the parameters were obtained was $\rho_e = 0.954$ and the measured normalized mean cycle time was 3.33 time units (the flow factor or X-factor). The actual performance and the approximate performance curve are depicted in Fig. 7. The number of servers and the cycle time offsets play important roles.

Table IV compares the normalized mean cycle time predictions for an $M/D/1$-queue, the standard failure prone $G/G/m$-queue approximation of (6) (including only the parallelism feature), and the extended approximation of (9) with the actual performance for the loading $\rho_e = 0.954$.

The prediction of (9) does not perform as well in this case as does the approximation of (6). However, it much more clearly demonstrates the components of the cycle time.

D. Chemical Vapor Deposition (CVD) Example

We consider the approximation based on one year of data from a chemical vapor deposition toolset (CVD). This class of CVD tool is a cluster tool with maximum achievable parallelism of $\phi = 1.70$. Again, we omit the details of the tool and time

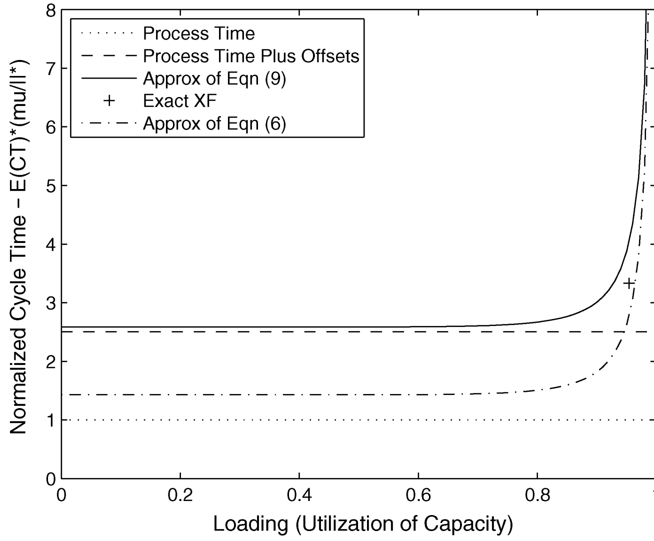

Fig. 7. A comparison of actual and predicted normalized cycle time (XF) in a 38-tool CMP toolset.

TABLE IV
NORMALIZED CYCLE TIME FOR CMP: 38 TOOLS

Model	$E(CT) * (\mu/\phi)$	$\|$% Error$\|$
$M/D/1$-queue	11.03	231%
Approx of Eqn (6) with m = 38	2.78	16.5%
(9) with m = 38	4.04	21.3%
Actual	3.33	-

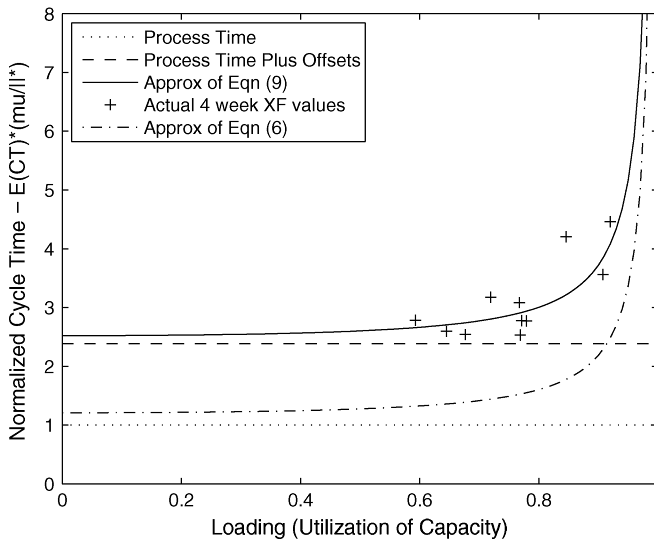

Fig. 8. One year of data, aggregated by month, is used to predict the normalized cycle time (XF) in a CVD toolset with parallelism and three servers.

frame. With the measured parameters, the (normalized) cycle time approximation of (9) becomes

$$E(CT) \approx (2.52) + (0.436)\frac{\rho_e^{1.83}}{3(1-\rho_e)}.$$

The actual performance for about one year aggregated every four weeks and the approximate performance curve are depicted in Fig. 8. For comparison, the approximation of (6), which includes only the parallelism complexity, is plotted as well. It appears that the new approximation performs well.

TABLE V
AVERAGE OF THE XF DATA POINTS FOR CVD

Overall XF Actual	3.23
Overall XF Prediction	3.14
% Difference	2.7%

An interesting issue arises when studying data collected from such a long time period. The loading during the study varied quite significantly from about 0.60 to 0.90. This reflects changes in the fabricator loading as discussed above. During each four week time frame, the approximation performs well (noise about the mean value is expected). Table V shows the overall actual mean cycle time (normalized and termed XF) for the entire time frame. The table also gives the overall predicted mean cycle time (normalized), which has been obtained by averaging the cycle time approximations we obtain for the loading at each data point (weighted by the number of lots, naturally). The resulting error is about 2.7%.

VIII. CONCLUDING REMARKS

By incorporating features found in practical manufacturing systems, we have suggested extensions to popular and intuitive closed-form approximations for the mean cycle time in $G/G/m$-queues. We tested the approximations using data obtained from IBM's 200 mm semiconductor wafer fabricator and found, not only that the extended approximations performed well, but that the model features incorporated played a significant role in the system performance.

It is important to note that the particular form of the loading term [e.g., $\rho^m/(1-\rho^m)$] used in the mean cycle time approximation is not important to our extensions. One could readily use others.

Many opportunities remain. Though we do not study them here, batch processing tools should be amenable to our approach. The model used to address production parallelism ignored interactions between lots with different production times (i.e., nonzero c_S^2) and nonideal availability. Further investigations should explore these issues. A more accurate approximation for idle with WIP would be obtained by separately measuring and treating the true increase in effective process time and the time that can be performed in parallel. Algorithms to deduce the effective number of servers should be investigated. A method for determining the tendency of lots to remain loyal or defect from a failed tool should be investigated as most tool sets will experience both. More rigor could be applied to the resulting approximation for defection. Investigations into the validity of assumptions such as the additivity of the cycle time offsets could be conducted. Networks of such tools should also be studied to assess the effect of the practical dynamics discussed.

ACKNOWLEDGMENT

The authors are deeply grateful to J. Fournier from IBM's Fab Operations Engineering Group, Vermont, for the extraction of important portions of the data presented. The authors are also grateful for the invaluable critique of the anonymous reviewers. This paper has benefited significantly from their suggestions.

REFERENCES

[1] J. F. C. Kingman, "The single server queue in heavy traffic," *Proc. Cambridge Philos. Soc.*, vol. 57, pp. 902–904, 1961.

[2] H. Sakasegawa, "An approximation formula $\mathrm{Lq} = \alpha\beta^{\rho}/(1-\rho)$," *Ann. Inst. Statist. Math.*, vol. 29, pp. 67–75, 1977.

[3] J. G. Shanthikumar and J. A. Buzacott, "On the approximations to the single-server queue," *Int. J. Prod. Res.*, vol. 18, pp. 761–773, 1980.

[4] W. Whitt, "Approximations for the GI/G/m queue," *Prod. Oper. Manage.*, vol. 2, no. 2, 1993.

[5] J. A. Buzacott and J. G. Shanthikumar, *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[6] W. J. Hopp and M. L. Spearman, *Factory Physics: Foundations of Manufacturing Management*, 2nd ed. New York: McGraw-Hill, 2001.

[7] L. Kleinrock, *Queueing Theory, Volume 1: Theory*. New York: Wiley, 1975.

[8] B. Avi-Itzhak and P. Naor, "Some queueing problems with the service station subject to breakdown," *Oper. Res.*, vol. 11, pp. 303–320, 1963.

[9] I. L. Mitrany and B. Avi-Itzhak, "A many-server queue with service interruptions," *Operations Research*, vol. 16, no. 3, pp. 628–638, 1969.

[10] P. J. Kuehn, "Approximate analysis of general queueing networks by decomposition," *IEEE Trans. Comm.*, vol. 27, pp. 113–126, 1979.

[11] W. Whitt, "The queueing network analyzer," *Bell Syst. Tech. J.*, vol. 62, no. 9, pp. 2279–2815, 1983.

[12] W. J. Hopp, M. L. Spearmana, S. Chayet, K. L. Donohue, and E. S. Gel, "Using and optimized queueing network model to support wafer fab design," *IIE Trans.*, vol. 34, pp. 119–130, 2002.

[13] D. P. Martin, "Capacity and cycle time—Throughput understanding system (CAC-TUS) an analysis tool to determine the components of capacity and cycle time in a semiconductor manufacturing line," in *Proc. 1999 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Boston, MA, Sep. 1999, pp. 127–131.

[14] K. Butler and J. Matthews, "How differentiating between utilization of effective availability and utilization of effective capacity leads to a better understanding of performance metrics," in *Proc. 2001 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Munich, Germany, 2001, pp. 21–22.

[15] K. Connerney, D. Martin, and R. Tomka, "Determining the capacity components of different classes of multichamber tools," in *Proc. 2001 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Munich, Germany, 2001, pp. 29–32.

[16] J. R. Morrison, B. S. Bortnick, and D. P. Martin, "Performance evaluation of serial photolithography clusters: Queueing models, throughput and workload sequencing," in *Proc. 2006 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Boston, MA, May 2006, pp. 44–49.

[17] J. van der Eerden, T. Saenger, W. Walbrick, H. Niesing, and R. Schuurhuis, "Litho area cycle time reduction in an advanced 300 mm semiconductor manufacturing line," in *Proc. 2006 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Boston, MA, May 2006, pp. 114–119.

[18] J. R. Morrison and D. P. Martin, "Performance evaluation of photolithography cluster tools: Queueing and throughput models," *OR Spectrum*, vol. 29, no. 3, pp. 375–389, 2007.

[19] J. H. Jacobs, L. F. P. Etman, E. J. J. van Campen, and J. E. Rooda, "Characterization of operational time variability using effective process times," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 3, pp. 511–520, Aug. 2003.

[20] J. H. Jacobs, P. P. van Bakel, L. F. P. Etman, and J. E. Rooda, "Quantifying variability of batching equipment using effective process times," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 269–275, May 2006.

[21] J. R. Morrison and D. P. Martin, "Approximate cycle time formulae for the G/G/m queue with server failures and constant cycle time offsets with applications," in *Proc. 2006 IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Boston, MA, May 2006, pp. 322–326.

[22] D. G. Kendall, "Some problems in the theory of queues," *J. Royal Statist. Soc. (B)*, vol. 13, pp. 151–173, 1951.

[23] S. L. Brumelle, "Some inequalities for parallel server queues," *Oper. Res.*, vol. 19, pp. 402–413, 1971.

[24] K. T. Marshall, "Some inequalities in queueing," *Oper. Res.*, vol. 16, pp. 651–665, 1978.

[25] S. Bose, *An Introduction to Queueing Systems*. Norwell, MA: Kluwer, 2001.

[26] H. White and L. Christie, "Queueing with preemptive priorities or with breakdown," *Oper. Res.*, vol. 6, pp. 79–95, 1958.

[27] D. P. Gaver, "A waiting line with interrupted service, including priorities," *J. Royal Statist. Soc. Series B*, vol. 24, pp. 73–90, 1962.

[28] J. Keilson, "Queues subject to service interruption," *Ann. Math. Statist.*, vol. 33, pp. 1314–1322, 1962.

[29] A. Federgruen and L. Green, "Queueing systems with service interruptions," *Oper. Res.*, vol. 34, no. 5, pp. 752–768, 1986.

[30] B. Avi-Itzhak, "A sequence of service stations with arbitrary input and regular service times," *Manage. Sci.*, vol. 11, no. 5, pp. 565,–571, 1965.

James R. Morrison (S'97–M'00) received the B.S. degree in electrical engineering and the B.S. degree in mathematics from the University of Maryland, College Park, in 1993, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Urbana, in 1997 and 2000, respectively.

He was with the Fab Operations Engineering Department , IBM Corporation, from 2000 to 2005. In 2005, he joined Central Michigan University, Mount Pleasant, as an Assistant Professor of Electrical Engineering. His research interests include semiconductor wafer fabrication, queueing networks, and stochastic control.

Donald P. Martin received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, and the M.Sc. (Eng.), DIC, and Ph.D. degrees in materials science from the Imperial College of Science and Technology, London, U.K.

He has managed a variety of engineering groups including manufacturing line support, technology development, process tool development, and new business introduction to manufacturing in IBM's Microelectronics Division. He presently works as a Senior Technical Staff Member in Industrial Engineering.