University of Würzburg
Institute of Computer Science
Research Report Series

Using Discrete–time Analysis in the Performance Evaluation of Manufacturing Systems

Mathias A. Dümmler[1] and Alexander K. Schömig[2]

Report No. 215 November 1998

[1] University of Würzburg
Department of Computer Science
Chair of Distributed Systems
Am Hubland, D-97074 Würzburg, Germany
E-Mail: *duemmler@informatik.uni-wuerzburg.de*

[2] Siemens AG
HL MS PR
D-81617 München, Germany
E-Mail: *alexander.schoemig@siemens-scg.com*

Published in: 1999 International Conference on Semiconductor Manufacturing Operational Modeling and Simulation (SMOMS '99), San Francisco, California, January 18-20, 1999.

The research for this paper was conducted while the second author was with the University of Würzburg, Department of Computer Science.

Using Discrete–time Analysis in the Performance Evaluation of Manufacturing Systems

Mathias A. Dümmler
University of Würzburg, Department of Computer Science, Chair of Distributed Systems, Am Hubland, D-97074 Würzburg, Germany, E-Mail: *duemmler@informatik.uni-wuerzburg.de*

Alexander K. Schömig
Siemens AG, HL MS PR, D-81617 München, Germany, E-Mail: *alexander.schoemig@siemens-scg.com*

Abstract

Discrete–time analysis (DTA) is a modelling technique that is mainly applied to the analysis of communication systems. In this paper, we show how DTA can be adopted to the numerical analysis of manufacturing systems. The modelling concepts of DTA are introduced and the advantages of DTA over other methods like discrete–event simulation are indicated. We apply DTA to two sample applications, namely the analysis of a simple batch server and a batch server with bounded idle time.

1 INTRODUCTION

Discrete–time analysis (DTA) techniques were first used to analyze basic single server systems. Ackroyd (1) shows how methods known from signal processing can be used to compute the waiting time distribution for the GI/G/1 queue. The signal processing algorithms used here apply, inherently, to *discrete–time* signals, and, hence, deal with discrete probability and density functions. However, these discrete–time methods can be used to model and analyze continuous time systems as well.

Tran-Gia (2) illustrates the use of DTA techniques for the performance evaluation and parameter estimation of ATM (asynchronous transfer mode) communication systems. For this type of queueing systems the discrete–time analysis technique is formidably suited since they basically have a discrete–time nature. Moreover, since cell blocking probabilities (which lie in the order of 10^{-5} to 10^{-9}) are in this environment the major performance measure of interest, discrete–event simulation requires excessive computing time.

In an attempt to provide a simple and computationally efficient method for the analysis of manufacturing systems, we have adopted the DTA approach to problems arising out of manufacturing. The algorithms and operators used in DTA have been implemented in a MATLAB toolkit, facilitating a simple realization of the analysis of complex systems using DTA.

In this paper, we first give an introduction to the modelling concepts of DTA. The discrete–time analysis approach is then applied to two sample applications, namely a simple batch server and a batch server with bounded idle time. The derivation of the analysis is provided in detail and numerical results are discussed. We conclude the paper by discussing the advantages and disadvantages of the discrete–time approach.

2 NOTATION IN DISCRETE–TIME ANALYSIS

Random variables representing periods of time considered in the course of this paper are of discrete–time nature, i.e., samples are multiples of Δt, the *unit length.* In other words, the time axis is divided into intervals of a fixed length Δt. Given a discrete–time random variable X, we denote its *distribution* (probability mass function) by

$$x(k) = \Pr\{X \;=\; k \cdot \Delta t\}\,, \quad k = \ldots, -2, -1, 0, 1, 2, \ldots\,. \tag{1}$$

The *mean* of a random variable X is defined as

$$\mathrm{E}[X] \;=\; \sum_{i=-\infty}^{\infty} i \cdot x(i)\,. \tag{2}$$

Given two independent random variables X_1 and X_2, their sum $X = X_1 + X_2$ is distributed according to the convolution of their individual distributions. Hence, the distribution $x(k)$ of X is

$$x(k) \;=\; \sum_{l=-\infty}^{\infty} x_1(l) \cdot x_2(k-l) \;=\; x_1(k) \circledast x_2(k)\,. \tag{3}$$

The symbol '$\circledast$' denotes the discrete convolution operation.

The operator $\pi_m()$ is used to "sweep up" all elements $x(k)$ of a distribution with $k < m$ and add their sum to the element $x(m)$. It is defined as

$$\pi_m(x(k)) \;=\; \begin{cases} 0 & , \quad k < m \\ \sum\limits_{i=-\infty}^{m} x(i) & , \quad k = m \\ x(k) & , \quad k > m\,. \end{cases} \tag{4}$$

Finally, we use the operator $\Delta_m()$ to shift the elements of a distribution down by m elements:

$$\Delta_m(x(k)) \;=\; x(k+m)\,. \tag{5}$$

3 SAMPLE APPLICATION I – A SIMPLE BATCH SERVER MODEL

Batch service or bulk service facilities, i.e., servers that can process a number of lots simultaneously, can be found in many areas of manufacturing. Implementations of batch servers also appear in transportation and distribution logistics. Examples of fabrication processes in semiconductor chip fabrication that are performed on batch service machines (furnace tubes) include the growth of thermal oxides, drive–in of dopants, glass reflow, and deposition of several types of material, e.g., polysilicon and silicon nitride (3). Notably, in all these cases additional jobs cannot be added to the service facility once service has been started, though there might be free capacity.

Some of these processes are very time-consuming (4). Therefore, on the one hand it may seem desirable always to load full batches. On the other hand, a furnace tube might have to wait a long time for lots to form a full batch, leaving this piece of equipment idle. This waiting time adds to the cycle time of a job and downstream stages face starvation. To find a fair trade–off between capacity utilization and cycle time one must study carefully the operating curves of a batch service facility running under different scheduling rules.

Neuts (5) presented an analytical study of batch service systems that operate according to a minimum batch size rule. In (6) *bulk queueing models* are discussed where jobs arrive in groups or are served as batches. Reviewing numerous papers a variety of single stage models is presented. Gold (7) investigates batch service systems in push and pull manufacturing environments using embedded Markov chain techniques. In (8) and (9) the problem of controlling batch processing systems in semiconductor wafer fabrication is approached from an operation manager's point of view.

3.1 Model description

In this section, we will consider the batch server model depicted in Figure 1. The model consists of a queue with infinite capacity and a batch server. The server has a maximum batch capacity of K lots. When a service period ends and there are less than L lots in queue, where $1 \leq L \leq K$ is an arbitrary constant, the server remains idle until L lots are accumulated in queue. Hence, the server is operated under a *minimum batch size* strategy.

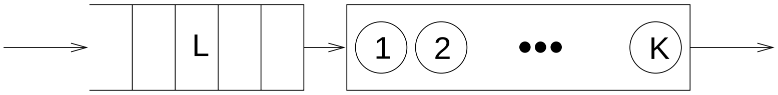

Figure 1: Simple batch server

The service time for the n–th batch is of length B_n, where the B_n, $n = 1, 2, \ldots$, are random variables following an arbitrary discrete–time distribution with mean $\mathrm{E}[B]$. Service times are independent of the batch size.

The interarrival times of lots follow a geometric distribution. To avoid batch arrivals, i.e., interarrival times of length zero, we shift the geometric distribution by one unit to the right. Hence, the random variable A_n, denoting the interarrival time between the $(n-1)$–th and the n–th lot follows the distribution

$$a_n(k) \quad = \quad \begin{cases} (1-p)^{k-1}\, p & , \quad k = 1, 2, \ldots \\ 0 & , \quad \text{else}\,. \end{cases} \tag{6}$$

where p is the probability of an arrival in one time unit. The mean interarrival time is $\mathrm{E}[A] = 1/p$. We define the system load ρ as the normalized offered traffic:

$$\rho \quad = \quad \frac{\mathrm{E}[B]}{K \cdot \mathrm{E}[A]}\,. \tag{7}$$

3.2 State distribution at departure instants

We now derive the steady state distribution of the number of lots in queue at departure instants using the embedded Markov chain technique. For an introduction to this technique, see (10).

Let X_{n-1} be the number of lots in queue immediately after the $(n-1)$–th batch has left the server. If $X_{n-1} \leq K$, the queue will be empty after the next service starts, since all lots waiting fit into a single batch. If there are more than K lots waiting, the number of lots in queue will be reduced by K. We denote the number of customers in queue immediately after the n–th service starts by Y_n. Hence we have

$$Y_n \quad = \quad \max(0, X_{n-1} - K)\,. \tag{8}$$

For the respective distribution $y_n(k)$ we get

$$y_n(k) \quad = \quad \pi_0(\Delta_K(x_{n-1}(k)))\,. \tag{9}$$

We will denote the number of arrivals during the n–th service as Γ_n. Since we assume geometrically distributed interarrival times, the distribution of the number of lots arriving during the n–th service period can be derived as

$$\gamma_n(k) \quad = \quad \sum_{m=k}^{\infty} \binom{m}{k} p^k (1-p)^{m-k}\, b_n(m)\,, \tag{10}$$

where $b_n(m)$ is the distribution of the length of the n–th service period.

Taking into account the number of arrivals during the n–th service, the number X_n of lots in queue immediately after the n–th service period is

$$X_n \quad = \quad Y_n + \Gamma_n\,, \tag{11}$$

and the respective distribution is

$$x_n(k) \quad = \quad y_n(k) \circledast \gamma_n(k)\,. \tag{12}$$

Using Equations 9 and 12, we obtain $x_n(k)$ from $x_{n-1}(k)$ as

$$x_n(k) \quad = \quad \pi_0(\Delta_K(x_{n-1}(k))) \circledast \gamma_n(k)\,. \tag{13}$$

We now iterate Equation 13 starting with an initial distribution $x_0(k)$, for example

$$x_0(k) \quad = \quad \begin{cases} 1 & k = 0 \\ 0 & k \neq 0\,, \end{cases} \tag{14}$$

which corresponds to an empty system. If the system is stable, i.e., $\rho < 1$, the series of distributions $x_n(k)$ converges to the steady state distribution of the number of lots in queue immediately after a service period ends:

$$x(k) = \lim_{n \to \infty} x_n(k)\,. \tag{15}$$

Note that by computing $x_n(k)$ for $n = 1, 2, \ldots$, we also obtain the transient behavior of the queueing system. I.e., we are able to observe the system state at arbitrary departure instants.

3.3 State distribution at arrival instants

We will now derive the steady state distribution $x^*(k)$ of the number of lots in queue observed at an arbitrary lot arrival instant. To this end, we consider the number of arrivals between two consecutive departures. The mean number of arrivals during an idle period is

$$\mathrm{E}[\Phi_{idle}] \quad = \quad \sum_{i=0}^{L-1} (L-i)x(i)\,, \tag{16}$$

and the mean number of arrivals during a busy period is

$$\mathrm{E}[\Phi_{busy}] \quad = \quad \frac{\mathrm{E}[B]}{\mathrm{E}[A]} \quad = \quad \rho \cdot K\,. \tag{17}$$

Hence, the mean number of arrivals between two consecutive departure instants is

$$\mathrm{E}[\Phi] \quad = \quad \mathrm{E}[\Phi_{idle}] + \mathrm{E}[\Phi_{busy}]\,. \tag{18}$$

Assume that the arrival instant occurs during a service period in steady state. It can be shown that the distribution $x^*_{busy}(k)$ of the number of lots in queue that a lot arriving during a service period observes is

$$x^*_{busy}(k) \quad = \quad \sum_{j=k+1}^{k+K} x(j)/\mathrm{E}[\Phi]\,. \tag{19}$$

We now assume that the lot arrival instant occurs during an idle period of the server. Assume further that there were j lots in queue at the previous departure instant, where $0 \leq j < L$. In this case, a lot arriving during an idle period will find $j, j+1, \ldots, L-1$ lots in queue with equal probability. Hence, the distribution of the number of lots in queue that a lot arriving during an idle period will encounter is

$$x^*_{idle}(k) \quad = \quad \sum_{j=0}^{k} x(j)/\mathrm{E}[\Phi]\,, \quad k = 0, 1, \ldots, L-1\,. \tag{20}$$

By combining x^*_{busy} and x^*_{idle}, we obtain the distribution of the number of lots in queue at an arbitrary arrival instant:

$$x^*(k) \quad = \quad x^*_{busy}(k) + x^*_{idle}(k)\,. \tag{21}$$

3.4 System characteristics

Using the steady state distributions at departure instants and at arrival instants, $x(k)$ and $x^*(k)$, we are able to derive a number of system characteristics of interest. The mean queue length is

$$\mathrm{E}[Q] \quad = \quad \sum_{i=1}^{\infty} i \cdot x^*(i)\,, \tag{22}$$

and the mean waiting time of lots before experiencing service is, according to *Little's Law* (cf. (10, p. 17)),

$$\mathrm{E}[W] \quad = \quad \mathrm{E}[Q] \cdot \mathrm{E}[A] \,. \tag{23}$$

Using Little's Law, we also obtain the mean number of lots in the server:

$$\mathrm{E}[L_{is}] \quad = \quad \mathrm{E}[B]/\mathrm{E}[A] \quad = \quad K \cdot \rho \,. \tag{24}$$

The second equality follows from Equation 7. Note that $\mathrm{E}[L_{is}]$ is a linear function of ρ and does not depend on the starting threshold L.

Finally, the mean number of lots per service is

$$\mathrm{E}[L_{ps}] \quad = \quad \sum_{i=0}^{L} L \cdot x(i) + \sum_{i=L+1}^{K} i \cdot x(i) + \sum_{i=K+1}^{\infty} K \cdot x(i) \,. \tag{25}$$

Note that in general $\mathrm{E}[L_{is}] \leq \mathrm{E}[L_{ps}]$, since the mean number of lots in server is averaged over both idle and busy periods, while the mean number of lots per start is averaged over busy periods only.

3.5 Numerical results

We consider a batch service system with maximum batch size $K = 4$ and deterministic service time of 30 time units. We vary the minimum batch size from $L = 1$, which corresponds to a greedy rule, to $L = 4$, corresponding to a full batch rule, in steps of 1. System load is $\rho = 0.1, \ldots, 0.9$.

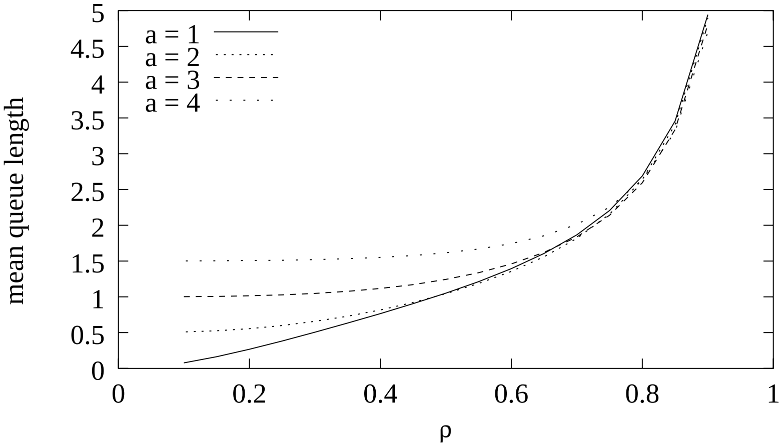

Figure 2: Mean queue length

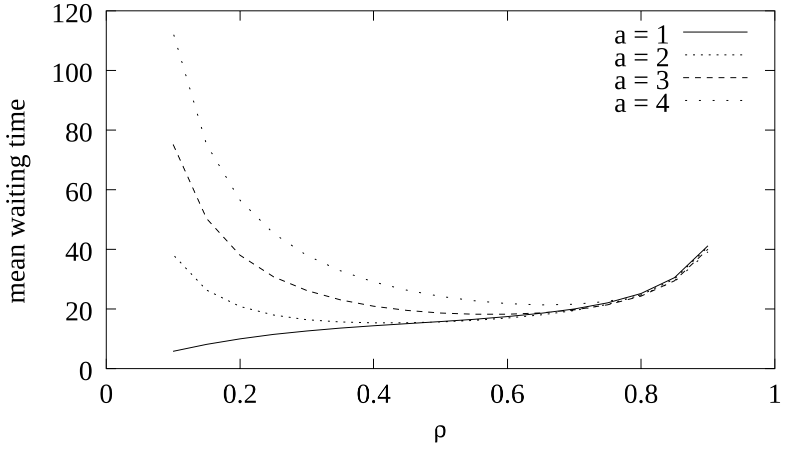

Figure 3: Mean waiting time

Due to lack of space we restrict the results to two system characteristics. When considering the mean queue length in Figure 2, we note that the starting threshold L has an influence on the queue length only if the load is relatively low. Obviously, the mean queue length can be reduced by choosing a lower starting threshold for a low system load ρ.

The mean waiting time depicted in Figure 3 is also smaller for low starting thresholds. As with the queue length, the influence of the starting threshold decreases when system load is increased. Note the bath tub-like shape of the waiting time curve, which is characteristic for batch service systems.

In Table 1, the mean waiting times gained from continuous–time discrete–event simulation are given. We compare them to the values derived with our proposed method. To show how decreasing the unit length Δt improves the accuracy of the results, we give the numerical results for values of $\Delta t = 1/2$ and $\Delta t = 1/10$.

Table 1: Simulation vs. discrete–time analysis: E[W]

	Simulation	Analysis			
		$\Delta t = 1/2$		$\Delta t = 1/10$	
ρ	E[W]	E[W]	error	E[W]	error
0.1	37.695 ± 0.508	38.003	0.82%	38.022	0.87%
0.2	20.499 ± 0.176	20.520	0.10%	20.578	0.38%
0.3	16.050 ± 0.098	15.988	0.39%	16.093	0.27%
0.4	15.017 ± 0.066	14.832	1.23%	14.992	0.17%
0.5	15.392 ± 0.076	15.145	1.61%	15.373	0.12%
0.6	16.845 ± 0.107	16.484	2.15%	16.814	0.19%
0.7	19.915 ± 0.242	19.157	3.81%	19.656	1.30%
0.8	26.264 ± 0.833	24.802	5.57%	25.637	2.39%
0.9	45.579 ± 4.015	42.122	7.58%	43.962	3.55%

For the simulation, the mean waiting time and the half width of the 95%–confidence interval are given. For the analytic results, the mean and the relative error are provided. An improvement in the accuracy of the results can be noted by decreasing Δt from 1/2 to 1/10. The computation times for this results were 373 seconds for the simulation and 19 seconds for the analysis on a Pentium II–266 processor, for both $\Delta t = 1/2$ and $\Delta t = 1/10$.

4 SAMPLE APPLICATION II – A BATCH SERVER WITH BOUNDED IDLE TIME

The batch server model considered in this section is very similar to the one in the previous section. In this model, however, we have added a timer. If the server becomes idle after a service period ends, the timer is started. If the idle time exceeds the maximum idle time t_o, service will be started with the lots waiting in queue, even if the number of lots waiting is smaller than the minimum batch size L. In case that the queue is empty at expiration of the maximum idle time, service will be initiated with the first lot arriving.

4.1 State distribution at departure instants

Extending the batch server model by the maximum idle time does not affect the derivation of the state distribution at departure instants presented in the previous section. To understand this, note that Equations 8 and 11 are still valid for the bounded idle time case. Hence, to compute the distribution $x(k)$ of the number of lots in queue at departure instants, we can apply Equation 13.

4.2 State distribution at arrival instants

The mean number of arrivals during an idle period is

$$\mathrm{E}[\Phi_{idle}] \;=\; \sum_{i=0}^{L-1} x(i) \cdot \sum_{j=1}^{\infty} \min(j, L-i) \cdot \tau(j) + x(0) \cdot \tau(0)\,, \tag{26}$$

where $\tau(k)$ is the distribution of the number of arrivals during the timeout interval. The mean number of arrivals during a busy period is the same as in Equation 17 of the previous model. The mean number of arrivals between two consecutive departure instants is $\mathrm{E}[\Phi] = \mathrm{E}[\Phi_{idle}] + \mathrm{E}[\Phi_{busy}]$.

Since the distribution of the number of lots during busy periods is not affected by the timeout period, the distribution $x^*_{busy}(k)$ of the number of lots in queue at an arrival instant falling into a service period can be gained as in Equation 19.

The computation of the distribution of the number of lots that an arrival will see during an idle period is best displayed in the form of an algorithm:

```
for all $i = 0, \ldots, L-1$ do
  $x^*_{idle}(i) := 0$
  for all $j = 1, \ldots, \infty$ do
    for all $k = 1, \ldots, j$ do
      $x^*_{idle}(i + \min(j, L-i) - 1) :=$
      $x^*_{idle}(i + \min(j, L-i) - 1) + x(i) \cdot \tau(j)/\mathrm{E}[\Phi]$
    end for
  end for
end for
$x^*_{idle}(0) := x^*_{idle}(0) + x(0) \cdot \tau(0)/\mathrm{E}[\Phi]$
```

We combine $x^*_{busy}(k)$ and $x^*_{idle}(k)$ as in Equation 21 to obtain the distribution of the number of lots in queue at an arbitrary arrival instant $x^*(\mathrm{k})$.

4.3 System characteristics

The mean queue length $\mathrm{E}[Q]$, the mean waiting time of lots $\mathrm{E}[W]$, and the mean number of lots in server $\mathrm{E}[L_{is}]$ can be derived according to Equations 22, 23, and 24, respectively. The mean number of lots per start is computed differently for this system:

$$\mathrm{E}[L_{ps}] = \sum_{i=0}^{L} \sum_{j=0}^{L-i-1} \max(i+j, 1) \cdot t(j) \cdot x(i) + \sum_{i=L+1}^{K} i \cdot x(i) + \sum_{i=K+1}^{\infty} K \cdot x(i) \,. \tag{27}$$

4.4 Numerical results

In Figures 4 and 5, the analytical results for a batch server with maximum idle time are displayed. The system parameters are as follows: Maximum batch capacity $K = 4$, starting threshold $L = 1, \ldots, 4$, deterministic service time $B = 30$ and maximum idle time of $t_o = 30$.

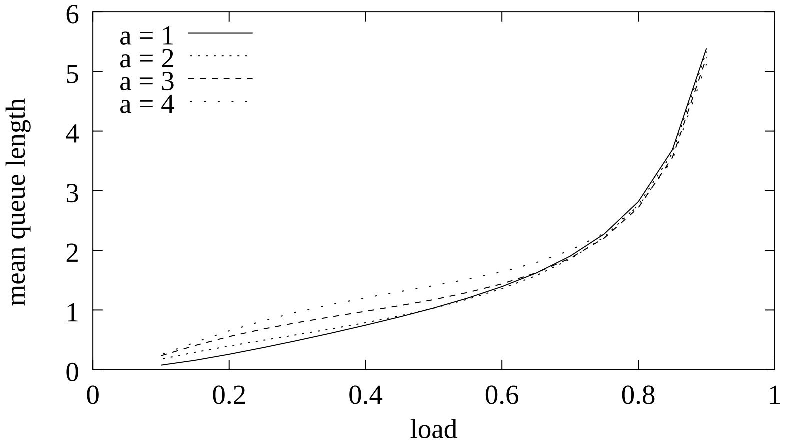

Figure 4: Mean queue length

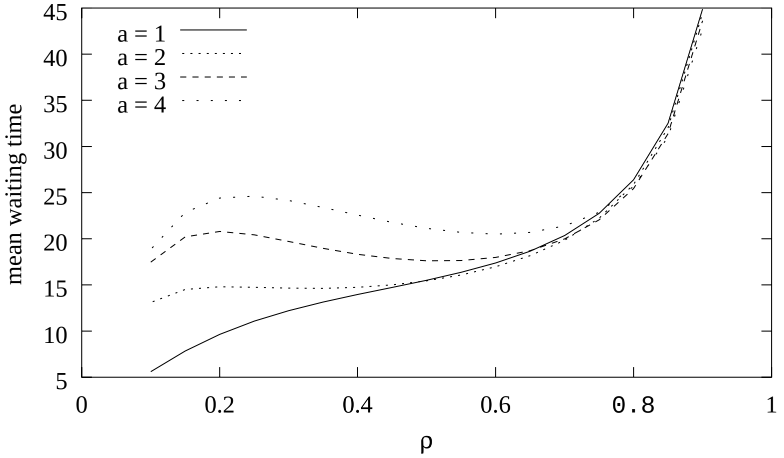

Figure 5: Mean waiting time

As we can see in Figure 4, the mean queue length varies less than in the simple batch server model for different values of L. More importantly, the mean waiting time in Figure 5 has an upper bound of about 25 for low values of ρ.

5 CONCLUSION

In this paper, we presented the discrete–time analysis (DTA) technique. The modelling concept of DTA and some operators were introduced. We applied DTA to the numerical analysis of two queueing systems, namely a batch service system operated under the minimum batch size rule and a batch service system with bounded idle time. We have implemented the analysis using a toolkit for MATLAB which provides all operators and algorithms necessary for DTA. The proposed method of analysis proved to be correct and computationally efficient. Using DTA in the area of semiconductor manufacturing allows to generate operating curves for a given machine or workstation with little effort. This facilitates the comparison of different operating strategies.

Comparing DTA to continuous–time discrete–event simulation, we note that the computation time for the analysis is much smaller than for the simulation. However, the applicability of DTA is restricted to a small part of a typical manufacturing system, for example a single workcell. This is due to the fact that the state space of the analytical model becomes very large when considering more complex systems. The advantages of DTA like little run time and memory usage can no longer be maintained for larger systems.

When modelling rare events like machine downtimes, excessive simulation times are necessary in discrete–event simulation to get meaningful results, whereas the computation time of the discrete–time analysis is hardly affected. To apply DTA, a certain amount of knowledge of probability and random processes is necessary. But this is also the case for simulation, since the simulation results are usually in the form of confidence intervals that require expert interpretation.

References

[1] M. H. Ackroyd, “Computing the waiting time distribution for the G/G/1 queue by signal processing methods,” *IEEE Transactions on Communication*, vol. COM-28, pp. 52–58, Jan. 1980.

[2] P. Tran-Gia, “Discrete–time analysis technique and application to usage parameter control modelling in ATM systems,” in *Proceedings of the 8th Australian Teletraffic Research Seminar*, (Melbourne, Australia), Dec. 1993.

[3] P. Singer, “Furnaces evolving to meet diverse thermal processing needs,” *Semiconductor International*, pp. 84–88, March 1997.

[4] P. K. Johri, “Practical issues in scheduling and dispatching in semiconductor wafer fabrication,” *Journal of Manufacturing Systems*, vol. 12, no. 6, pp. 474–485, 1992.

[5] M. F. Neuts, “A general class of bulk queues with poisson input,” *Ann. Math. Stat.*, vol. 38, pp. 759–770, 1967.

[6] M. L. Chaudhry and J. G. C. Templeton, *A First Course in Bulk Queues*. New York: Wiley, 1983.

[7] H. Gold, *Performance Modelling of Batch Service Systems with Push and Pull manufacturing Management Policies*. PhD thesis, Universität Würzburg, Fakultät für Mathematik und Informatik, 1992.

[8] C. R. Glassey and W. W. Weng, “Dynamic batching heuristics for simultaneous processing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 4, pp. 77–82, May 1991.

[9] J. W. Fowler, D. T. Phillips, and G. L. Hogg, “Real-time control of multiproduct bulk-service semiconductor manufacturing processes,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 5, pp. 158–163, May 1992.

[10] L. Kleinrock, *Queueing Systems, Vol. 1: Theory*. New York: Wiley, 1975.