Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

ON USING CONTINUOUS FLOW LINES FOR PERFORMANCE ESTIMATION OF DISCRETE PRODUCTION LINES

Rajan Suri
Bor-Ruey Fu

Department of Industrial Engineering
University of Wisconsin – Madison
1513 University Avenue
Madison, Wisconsin 53706

ABSTRACT

We explore the use of models of *continuous tandem* (CT) lines for performance analysis of *discrete tandem* (DT) production lines. We formalize the translation of input parameters and performance measures (PMs) between DT and CT lines. We show that the CT model can be represented as a generalized semi-Markov process (GSMP). This leads to a concise simulation algorithm for a CT model. We show empirically that the CT model provides reasonable estimates for the DT line PMs. We provide preliminary results on gradient estimation for CT models via infinitesimal perturbation analysis (IPA). The aim of the paper is to provide a basis for the further exploration of CT models as a means to parameter optimization for DT lines.

1 INTRODUCTION

There exist two distinct types of manufacturing systems: *discrete* and *continuous*. These terms indicate whether material moves through the processes as discrete entities (e.g. an automobile factory) or as continuous fluid (e.g. a chemicals plant). In this paper we are concerned with a particular configuration of production line, namely a *tandem* line consisting of a sequence of machines in series (Figure 1).

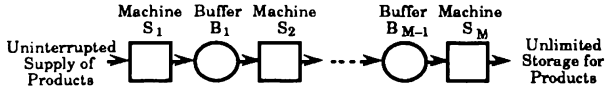


Figure 1: Tandem Production Line

This paper summarizes some initial work which is part of our long term research aim to enhance the set of tools available for modeling and parameter optimization of *discrete tandem* (DT) lines. Further details are available in Suri and Fu (1991a and 1991b).

Our approach focuses on the analysis of DT lines using *continuous tandem* (CT) models. The reasons why it is worth exploring this avenue are as follows. i) the optimization of real world DT lines continues to require large amounts of computational effort (e.g. see Wei, Tsao, and Otto 1989). We feel the simulation algorithm and other characteristics of the CT model (below) will provide considerable increase in computational efficiency. ii) all the parameters in the CT model are continuous (e.g. buffer sizes). For stochastic systems, techniques for continuous parameter optimization are much more efficient, and better understood than those for discrete parameter optimization (Glynn 1986). iii) with continuous parameters there is the possibility of using gradient information to speed up the optimization algorithm. Recently there have been many advances in gradient estimation for stochastic systems (see Suri 1989), and there is much evidence for the fact that such estimates considerably improve the convergence rate of optimization algorithms. iv) a few previous researchers have also considered the use of CT models to analyze DT lines (see section 1.2). However, this has been done on a somewhat *ad-hoc* basis, with each study using different assumptions. We hope to provide a formal framework.

Our research concept is summarized in Figure 2. The figure indicates that production lines are currently analyzed using simulations of DT lines (A). The first step is to study the relation between DT and CT models (B), and then to obtain methods for optimizing CT models. With such tools available, we could then hope to use CT models themselves as decision support tools for production lines (C). Figure 3 outlines the main research steps needed to obtain this goal.

There is a substantial body of literature on the analysis of tandem production lines (see the bibliography in Suri, Sanders, and Kamath 1992). Even so, analytic results are available only for 2-machine and 3-machine lines. The analysis of longer lines has involved the use of heuristic approximations or simulation models.

Analytic solutions for 2-machine DT lines are given in Gershwin and Schick (1983). For longer DT lines, approximate solution algorithms are proposed in Choong and Gershwin (1987), and Dallery, David, and Xie (1988). An alternative approach to DT line analysis has been to use simulation along with gradient estimation and optimization techniques (Ho, Eyler, and Chien 1983).

Solutions for 2-machine CT lines are in Gershwin and Schick (1980), and Dallery, David, and Xie (1989). An approximation for longer CT lines is presented in Dallery, David, and Xie (1989). A comparison of 2-machine DT and CT models can be found in Koster and Wijngaard (1989) and for longer lines in Alvarez, Dallery, and David (1991). These results indicate that the use of a CT model to approximate a DT line is often justified.

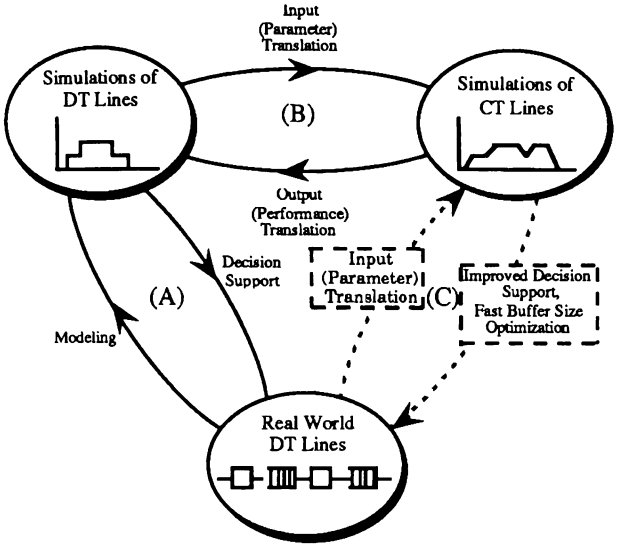


Figure 2: Overview of Research Concept

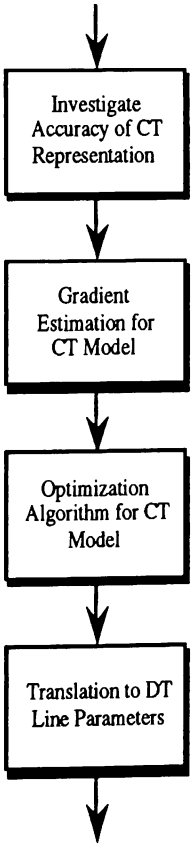


Figure 3: Outline of Ongoing Research Steps

In our search of the literature, we have not found previous work that provides a clear translation of parameters and detailed PMs between the two models, and which thoroughly compares the detailed PMs for the cases of longer lines. There also does not appear to be any work on gradient estimation or optimization for simulations of CT lines.

2 MODEL DESCRIPTION

A tandem line, whether discrete or continuous, consists of M processing machines (S_1, ..., S_M) in series, connected by $M-1$ buffers (B_1, ..., B_{M-1}), see Figure 1. A line in which the cycle times at all machines are the same is called a *homogeneous* line. Here we will allow the cycle times to be different (a *nonhomogeneous* line).

We discuss a short sample path for a 2-machine system for each of the two models. The two sample paths are shown in Figure 4. First we discuss the DT line, see Figure 4(a). Let the cycle time for S_1 be *1* and for S_2 be *2*, and the capacity of B_1 be *1*. The products being made are labelled as *P1*, *P2*, and so on. Assume that, at time *0*, both S_1 and S_2 are working (not failed), with *P2* starting a cycle at S_1, and *P1* at S_2, and that B_1 is empty. At time *3*, *P4* completes its cycle at S_1 but since B_1 is full, *P4* cannot leave, and S_1 cannot continue to work. In this case, S_1 is said to be *blocked*. Similarly, the reader can follow the sample path through time *6*. Suppose the first failure of S_1 is destined to occur after it has operated for *4.5* time units. This is called the *operating time to failure*. Then we see in Figure 4(a) that S_1 fails at time *6.5* (since it was not working for 2 time units while it was blocked), while still processing *P6*. S_1 is now said to be *down*. Meanwhile, S_2 continues to operate, but at time 10 we see that it has no more products to work on. It is said to be *starved*. Suppose S_1 is repaired *4.5* time units after it failed. This is called the *time to repair*. Then it starts working again at time *11*, finishes its work on *P6* at time *11.5*, at which point *P6* goes immediately to S_2 which can now start working again.

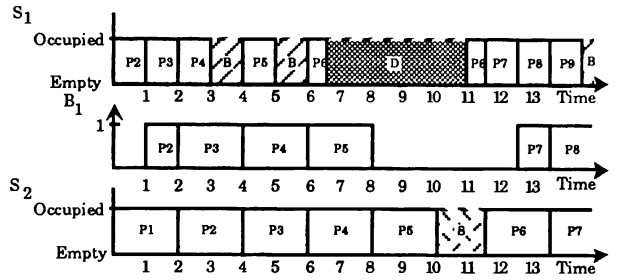
Figure 4(b) shows a sample path for a CT line. We have chosen parameters and operating conditions in a way such that the sample path "resembles" that of the DT line. The processing capacity of S_1 is *1* unit of volume per unit time, and that of S_2 is *0.5*. The capacity of B_1 is *1* as before. Let v_1 and v_2 be the actual processing rate of S_1 and S_2, at a given point in time. Figure 4(b) shows plots of v_1 and v_2 over time. Also shown is the level of the buffer. We start at time *0* with the buffer empty and S_1 working at its full rate of *1*. As soon as S_1 begins producing, due to the nature of the CT line, product is immediately available to S_2 which begins working at its full rate of *0.5*. Since the rate of S_2 is less than that of S_1, the buffer level starts to rise at the

rate of *0.5* units per unit of time. At time *2*, the buffer is full, so S_1 is constricted by S_2 and the processing rate at S_1 is reduced to *0.5*. Note, however, that unlike in the discrete case where S_1 got blocked and completely stopped processing, in this case product still keeps flowing through S_1, B_1, and S_2. We will say S_1 is in a *reduced flow* condition. There is a strong correspondence between the two sample paths. Consider time units *2* through *6* for S_1 in both lines. We see that the *average* production rate in both cases is *0.5*. The reduced flow in the CT line thus mimics, in a "smoothed out" manner, the blocking in the DT line.

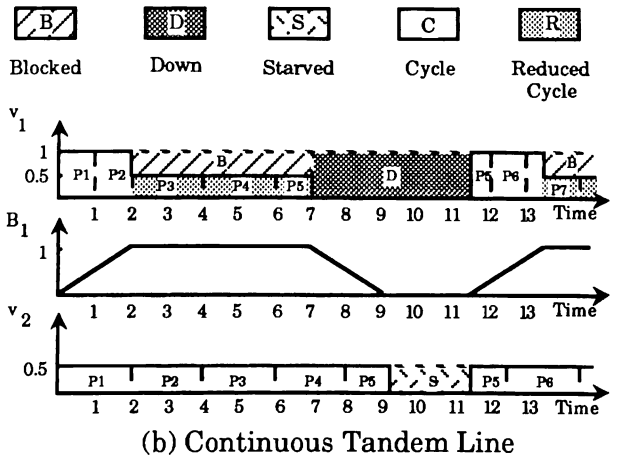
Next, for the CT line we suppose that the failure of S_1 occurs after it produces *4.5* units of volume. We will call this the *operating volume to failure*. Because of the periods of reduced flow rate of S_1, this failure will occur at time *7*. S_2 can still keep processing and the buffer level decreases at a rate of *0.5*. At time *9* the buffer becomes empty and S_2 is *starved*. As before, we suppose the repair of S_1 occurs *4.5* time units after the failure, so in this case S_1 is repaired at time *11.5*. Note that S_2 also resumes its operation instantaneously.

These observations will form the core of our method for translating performance measures from the CT model to the DT line.

Now we formalize our models for the two types of lines. The following assumptions apply to both types of models.



(a) Discrete Tandem Line



(b) Continuous Tandem Line

Figure 4: Sample Paths of 2-Machine Lines

A1. There is an unlimited supply of material available to S_1, that is, S_1 is never starved.

A2. There is an unlimited storage area following S_M, that is, S_M is never blocked.

A3. There is no transfer delay.

A4. A machine can fail only when it is operational.

A5. The repair time for each machine is exponentially distributed. (A general distribution can be used here. However, we choose the exponential distribution in order to compare our results with analytic ones, where available.)

For the DT model, we have the following additional assumptions.

A6D. The cycle times of machines are deterministic.

A7D. The operating time to failure for each machine is exponentially distributed. (The remark in A5 applies here too.)

A8D. Machines operate and are blocked via the "manufacturing blocking" procedure (Altiok and Stidham 1982).

With the above assumptions, our DT line model can be completely characterized by the following parameters:

T_i Cycle time of S_i

f_i Mean operating time between failures for S_i

r_i Repair rate for S_i

$B(j)$ Buffer size of B_j (non-negative integer)

For the CT model, we have the following additional assumptions.

A6C. The processing rate at a machine can range from *0* to its maximum processing capacity.

A7C. The operating volume to failure for each machine is exponentially distributed. (The remark in A5 applies here too.)

With the above assumptions, our CT line model can be completely characterized by the following parameters:

C_i Maximum flow rate (processing capacity) of S_i

w_i Mean operating volume to failure for S_i

r_i Repair rate for S_i

$B(j)$ Buffer size of B_j (non-negative real number)

We propose the following translation for the parameters from the DT model to the CT model:

$C_i = 1/T_i$

$w_i = f_i/T_i$

r_i and $B(j)$ remain the same.

3 A FORMAL MODEL FOR A CT LINE

Now we provide a formal model for the dynamics of a CT line. For DT lines, such a model is commonly available, thus we will only present a CT line model here.

A generalized semi-Markov process (GSMP) is a mathematical framework for studying DEDS. For more about GSMPs see Glynn (1989).

Now we show that it is possible to construct a GSMP representation for the dynamics of a CT line. This may be surprising since one associates GSMP models with DEDS which typically have discrete entities (e.g. customers), and as stated by Glynn (1989), "DEDS are fre-

quently used as models of systems having piecewise constant trajectories". Thus, one does not expect a model with continuous fluid in it, and with a time-varying trajectory as in Figure 4(b), to be representable as a GSMP. Making this formal connection has some benefits: i) it provides a formal model of CT lines that might serve as a standard reference for researchers. Such a standard seems to be lacking in the CT line literature. ii) the resulting simulation algorithm is simple. iii) many results are available in the context of GSMPs and these could be brought to bear on CT models where needed (Glynn 1989). Also Glasserman (1991) provides several results on consistency of IPA for gradient estimation in GSMPs.

First we define the physical state of the line. The discussion here is greatly abbreviated (see Suri and Fu 1991a for details). Let t denote time and $\alpha_i(t)$ be the state of S_i at time t, where

$$\alpha_i(t) = \begin{cases} \boldsymbol{D}, & \text{if } S_i \text{ is down (failed),} \\ \boldsymbol{O}, & \text{if } S_i \text{ is operational (full flow),} \\ \boldsymbol{S}, & \text{if } S_i \text{ is starved (reduced/zero flow),} \\ \boldsymbol{B}, & \text{if } S_i \text{ is blocked (reduced/zero flow).} \end{cases}$$

Let $v_i(t)$ be the flow rate of S_i at time t. Let $x_j(t)$ be the level of B_j at time t.

The set of event types in a CT line is $E = \{\mathcal{F}_i, \mathcal{R}_i, \mathcal{BE}_j, \mathcal{BF}_j, \mathcal{T}_f: i = 1, 2, ..., M, j = 1, 2, ..., M-1\}$, and each of these events is defined in Table 1.

Table 1: Summary of Events

Symbol	Event Represented
$\mathcal{F}_i$	Failure of S_i
$\mathcal{R}_i$	Repair of S_i
$\mathcal{BE}_j$	B_j becomes Empty
$\mathcal{BF}_j$	B_j becomes Full
$\mathcal{T}_f$	Termination of the simulation

In a GSMP, each event is associated with a clock representing the residual lifetime of that event, and each clock has a speed at which it runs down. When the clock associated with an event runs down to 0, that event occurs. Upon the occurrence of an event, changes may occur to the physical state, clock settings, and clock speeds, according to defined rules. The clock definitions and speeds for the CT line are in Table 2.

Now we describe how the system evolves. Let

$k(e, t)$ be the reading of the clock for event e at time t
$r(e, t)$ be the speed of $k(e, t)$
$E(t)$ be the set of events for which $r(e, t) \neq 0$

$E(t)$ can be thought of as the current set of possible events. At time t, the additional time to the possible occurrence of event e is defined as $\Delta t(e, t) = k(e, t)/r(e, t)$, for $e \in E(t)$.

With the above definition, the next event to occur is given by

$$e^*(t) = \underset{e}{argmin}\ \{\Delta t(e, t): e \in E(t)\}.$$

Now consider the effect of an event, say e, on the system. Define pseudo-buffers B_0 and B_M with $x_0(t) = \infty$ (for all t) and $B(M) = \infty$. We start by understanding the "local" dynamics of a CT line, i.e. how buffers and machines interact with their immediate upstream and downstream neighbors:

If event at time t is $\mathcal{BF}_i$ then
  $\{ v_i(t) = v_{i+1}(t);\ \alpha_i(t) = \boldsymbol{B}; \}$
If event at time t is $\mathcal{BE}_i$ then
  $\{ v_{i+1}(t) = v_i(t);\ \alpha_{i+1}(t) = \boldsymbol{S}; \}$
If event at time t is $\mathcal{F}_i$ then
  $\{ v_i(t) = 0;\ \alpha_i(t) = \boldsymbol{D};\ k(\mathcal{R}_i, t) = sample(1/r_i); \}$
If event at time t is $\mathcal{R}_i$ then $\{$
  set $v_i(t)$ according to Table 3;

$$\alpha_i(t) = \begin{cases} \boldsymbol{S}, & \text{if } x_{i-1}(t) = 0 \text{ and } v_i(t) = v_{i-1}(t), \\ \boldsymbol{B}, & \text{if } x_i(t) = B(i) \text{ and } v_i(t) = v_{i+1}(t), \\ \boldsymbol{O}, & \text{otherwise;} \end{cases}$$

  $k(\mathcal{F}_i, t) = sample(w_i); \}$

where $sample(\mu)$ denotes a sample taken from an exponential distribution with mean μ.

Finally we model the "global" dynamics, i.e., how an event at S_i or B_i affects buffers and machines in the rest of the line. Here we simply present the logic for the upstream and downstream effects of a change. For a detailed explanation see Suri and Fu (1991a).

If $v_i(t)$ is changed at time t then do $\{$
  $m = i;$
  while $(x_m(t) = 0)$ and $(m \leq M - 1)$ do $\{$
    $v_{m+1}(t) = min(v_m(t), C_{m+1});$
    if $v_{m+1}(t) = v_m(t)$ then $\alpha_{m+1}(t) = \boldsymbol{S};$
    else $\alpha_{m+1}(t) = \boldsymbol{O};$
    $m = m + 1; \}$
  $m = i - 1;$
  while $(x_m(t) = B(m))$ and $(m \geq 1)$ do $\{$
    $v_m(t) = min(C_m, v_{m+1}(t));$
    if $v_m(t) = v_{m+1}(t)$ then $\alpha_{m+1}(t) = \boldsymbol{B};$
    else $\alpha_m(t) = \boldsymbol{O};$
    $m = m - 1; \} \}$

Any clock, clock speed, or other physical variable not updated in the above equations simply retains its value prior to the event. After this event, the system proceeds along its sample path until the next event occurs, and the process repeats itself until the termination event.

This completes the formal description of the CT model in terms of a GSMP. A simulation algorithm implementing the above logic is given in the Appendix.

4 TRANSLATION OF PERFORMANCE MEASURES

We assume that the PMs to be estimated for the DT line are steady state values of: line throughput, average buffer levels, machine utilizations, average product flow time,

Table 2: Clock Definitions and Clock Speeds

Symbol	Clock Description	Clock Speed†
$W_i(t)$	Remaining operating volume to failure for S_i at time t	$v_i(t)$
$U_i(t)$	Remaining repair time for S_i at time t	$\mathbf{1}_{\{\alpha_i(t) = D\}}$
$x_j(t)$	Level of B_j at time t	$[v_{j+1}(t) - v_j(t)]^+$
$B(j) - x_j(t)$	Excess capacity of B_j at time t	$[v_j(t) - v_{j+1}(t)]^+$
$q(t)$	Remaining volume to be produced by S_M at time t	$v_M(t)$

†$\mathbf{1}_{\{\cdot\}}$ is the indicator function of set $\{\cdot\}$. $[x]^+ = max(0, x)$.

Table 3: Flow Rate Update Table for S_i

If		then
$x_{i-1}(t)$	$x_i(t)$	$v_i(t)$
$=0$	$=B(i)$	$min\ (v_{i-1}(t), C_i, v_{i+1}(t))$
$=0$	$<B(i)$	$min\ (v_{i-1}(t), C_i)$
>0	$=B(i)$	$min\ (C_i, v_{i+1}(t))$
>0	$<B(i)$	C_i

and average work-in-process (WIP). The definitions of these PMs for DT lines are well accepted. On the other hand, for CT lines it will be seen that the definitions of certain PMs are not straightforward. While several definitions could be justified, in each case we propose one that will most closely track the DT line PM.

Let $t_0, t_1, \ldots$ be the times of the *0th, 1st, ...* occurrences of events in a sample path. We assume that standard methods are being used to estimate steady state PMs from a finite observation period, namely the interval from t_m to t_n (where $t_m < t_n$). Define $\Delta t_i = t_i - t_{i-1}$ and $T = t_n - t_m$. For the derivations below, note that the machine states do not change in $[t_{i-1}, t_i)$ and hence the flow rate of machines is piecewise constant.

We use the argument "DT" with a PM to denote the definition for a DT line. PMs without this argument denote our suggested definitions of PMs derived from the CT line model.

Throughput: For a DT line, the throughput is defined by TP_{DT} = no. of pieces produced by S_M in $[t_m, t_n]/T$. For the CT line, we define throughput in the natural way, namely,

$$TP = \frac{1}{T}\int_{t_m}^{t_n} v_M(t)\,dt = \frac{1}{T}\sum_{i=m+1}^{n} v_M(t_{i-1})\cdot\Delta t_i,$$

where we use the fact that $v_M(t)$ is piecewise constant.

Average buffer level: For the CT line we define average buffer level in the same way as for a DT line:

$$\bar{x}_j = \frac{1}{T}\int_{t_m}^{t_n} x_j(t)\,dt.$$

Machine utilizations: In a DT line, the utilization of a machine is broken out into four categories: cycle, starved, blocked, and down. Each represent the proportion of time the machine spends in the corresponding state.

For a machine in the DT line, the cycle utilization represents the proportion of time the machine is busy. This is not the quantity we should observe for the CT line, since a machine can be busy but working at a highly reduced rate. To get to a suitable definition for the CT line, consider again the DT line definition:

$$U_{c,\,DT}(i) = \text{time in cycle/total time} = (\text{time in cycle}/T_i)/(\text{total time}/T_i).$$

Now the numerator is the number of pieces actually produced in the observation period, while the denominator represents the potential number of pieces that could have been produced. With this in mind we propose:

$$U_c(i) = \frac{1}{C_i T}\int_{t_m}^{t_n} v_i(t)\,dt = \frac{1}{C_i T}\sum_{k=m+1}^{n} v_i(t_{k-1})\cdot\Delta t_k.$$

Similarly, we propose for the starved utilization and blocked utilization:

$$U_s(i) = \frac{1}{C_i T}\sum_{k=m+1}^{n} [C_i - v_i(t_{k-1})]\cdot\mathbf{1}_{\{\alpha_i(t_{k-1}) = S\}}\cdot\Delta t_k.$$

$$U_b(i) = \frac{1}{C_i T}\sum_{k=m+1}^{n} [C_i - v_i(t_{k-1})]\cdot\mathbf{1}_{\{\alpha_i(t_{k-1}) = B\}}\cdot\Delta t_k.$$

The indicator function ensures the integral takes into account only the times when S_i is starved (or blocked).

The down utilization is

$$U_d(i) = \frac{1}{T}\sum_{k=m+1}^{n} \mathbf{1}_{\{\alpha_i(t_{k-1}) = D\}}\cdot\Delta t_k.$$

Average WIP: In a DT line, this includes products at the buffers and at the machines. In our CT model, however, there is *no* WIP at a machine. Thus we need to account for this difference. In a DT line the average WIP in a machine S_i is given by $1 - U_S(i)$. Since we already have an estimate for the DT line value of $U_S(i)$ from our CT model, we can use this to get an estimate of the average WIP for the DT line by adding a term to the CT line WIP, giving

$$WIP = \sum_{j=1}^{M-1} \bar{x}_j + \sum_{i=1}^{M} [1 - U_S(i)].$$

Average Flow Time: Since the material in a CT model is continuous fluid the calculation of product flow time would require complex integration to account for the time spent in the system by each infinitesimal piece of material. Rather than compute this, we suggest using an indirect estimate of the average flow time via Little's law. Therefore, we define a *pseudo* flow time as Flow Time = *WIP/TP*.

This completes our set of proposed definitions of PMs for the CT line. The merits of all these proposed definitions will be tested via numerical results.

5 NUMERICAL RESULTS

Now we compare the PMs estimated from simulations of DT lines (DTL) and CT lines (CTL). Our simulation program for CTL, written in the C language, is an implementation of the algorithm in the Appendix, while the simulation of the DTL is written in the SIMAN language. The PMs shown in the tables below are obtained from the average of *10* independent replications. For each replication, the simulation is run for *10,000* time units (warm up) to eliminate the effects of the initial transient, and then statistics are collected for a production quantity of *30,000*.

Three cases of CT lines are considered. Case 1 is a 2-machine homogeneous line and Case 2 is a 2-machine non-homogeneous line, both from Gershwin and Schick (1980). Case 3 is a 6-machine non-homogeneous line from Dallery, David, and Xie (1989). The input parameters for the CT lines are listed in Tables 4 and 5. The last column lists the isolated throughput (TP_i) of S_i defined as $TP_i = w_i/(w_i/C_i + 1/r_i)$. The isolated throughput gives an indication of how well the machines are matched.

For Cases 1, 2, and 3, we translated the parameters to their DT line equivalents (section 2) and simulated the resulting DTL. Table 6 compares the main PMs from simulations of DTL and CTL for all three cases, along with 95% confidence intervals. The last column shows the percentage deviation of the average CT line simulation estimate from the average DT line simulation estimate. We see that the throughput and WIP estimates are within 3%, while the average buffer level estimates are within 4%.

Tables 7, 8, and 9 show the details of machine utilizations for Cases 1, 2, and 3, respectively. (such detailed PMs have not been defined and studied in previous work.) For Case 1 (Table 7), all the machine utilizations of the CT line agree with those of the DT line to within 10^{-3}. The estimates of starved utilization of S_2 are not significantly different from zero ($< 2\times10^{-4}$) in both the DT and CT lines, so an error term is not calculated here.

From Tables 8 and 9 we see that for Cases 2 and 3 the relative errors of the detailed machine utilizations for blocked, cycle and down are within 8%. The errors for starved utilizations are higher. We will now discuss this further.

Of course, there are fundamental differences in the operation of CT and DT lines. Beyond the simple fact that entities are discrete in one case and continuous in another, is the fact that this affects the detailed dynamics of the line. For instance, suppose S_i is starved because of the failure of S_{i-1}. In a DT line, after S_{i-1} is repaired, S_i still has to wait for a workpiece to complete its processing at S_{i-1} before S_i can start working again. On the other hand, in a CT line, as soon as S_{i-1} is repaired it starts producing, and S_i is also *immediately* operational. These (and other) differences in the dynamics mean that we can certainly expect some discrepancies between the detailed performance measures of the lines. While the detailed utilizations of the machines are not as accurate, we find the main PMs (throughput, WIP) are still surprisingly accurate. Furthermore, as shown in Suri and Fu (1991a) the CTL estimates typically require less computation time since fewer events are processed.

6 GRADIENT ESTIMATION

In this section we preview some of our results on gradient estimation for CT lines. The aim is to show that, although the IPA technique for gradient estimation was originally developed for systems with discrete entities, it can be applied (with appropriate extensions) to CT lines as well. We consider estimating the gradient of steady state throughput with respect to the flow rates of the M machines (i.e. the M-vector of values dTP/dC_m, for $m = 1, \ldots, M$). The IPA algorithm for calculating this gradient vector is concise and easily inserted in the simulation code, as seen in the Appendix. The derivation of

Table 4: Input Parameters for Case 1 and 2

Case	M/c S_i	Flow Rate C_i	Mean Operating Volume to Failure w_i	Repair Rate r_i	Buffer Size $B(j)$	Isolated Throughput TP_i
1	1	1	10	0.7	4	0.875
	2	1	1.429	0.1		0.125
2	1	3	2.5	0.4	5	0.75
	2	5	2.976	0.32		0.80

Table 5: Input Parameters for Case 3

M/c S_i	Flow Rate C_i	Mean Operating Volume to Failure w_i	Repair Rate r_i	Buffer Size $B(i)$	Isolated Throughput TP_i
1	2.809	25.955	1.316	4	2.596
2	3.571	160.714	0.2	2	3.214
3	3.571	160.714	0.2	2	3.214
4	3.571	160.714	0.2	2	3.214
5	3.571	160.714	0.2	4	3.214
6	2.882	26.628	1.316		2.663

the algorithm will not be discussed here -- see Suri and Fu (1991b) for a full discussion.

Preliminary numerical results indicate that this algorithm is accurate. As a first example, we compare the estimates from the IPA algorithm with those obtained by a central finite difference (CFD) estimate from the analytical formula for 2-machine lines (denoted "CFD+formula"). This is done for two lines, namely Cases 1 and 2 in section 4. (We chose the finite difference $\delta = \pm 0.01 C_m$ to obtain CFD estimates for all the tables in this section). The results are in Table 10. We see that the IPA algorithm appears to give accurate estimates for the four gradients shown in the Table.

Next we compare the IPA estimate with a CFD estimate obtained via multiple simulation runs for a 6-machine non-homogeneous line taken from Dallery, David, and Xie (1989). The input parameters for this line (Case 4) are in Table 11.

The numerical results are in Table 12. The IPA gradient vector estimate is obtained from observing *10* independent replications. For each replication, the simulation is run for a quantity of *100,000* products (warmup) to eliminate the effect of the initial transient, and then statistics are collected for a production quantity of *5,000,000*. On the other hand, to get n gradient estimates using CFD we need $2n$ sets of replications, each set having the run lengths just stated. The gradient estimates are shown in the table along with their *95%* confident intervals. It can be seen that the IPA estimate compares well with the "CFD+simulation" estimate. However, the CFD+simulation estimate takes 6.6 times the computational effort of the IPA estimate. (The CPU times are for a Sun SPARC workstation).

Table 6: Main Performance Measures

	CTL Simulation	DTL Simulation	Error
Case 1	0.125±0.001	0.125±0.001	0.0%
TP Case 2	0.561±0.003	0.573±0.004	-2.1%
Case 3	2.112±0.016	2.141±0.018	-1.4%
Case 1	5.970±0.001	5.979±0.001	-0.2%
WIP Case 2	4.035±0.017	4.041±0.033	-0.1%
Case 3	11.229±0.088	11.375±0.330	-1.3%
Avg. Buffer			
Case 1	3.971±0.001	3.980±0.001	-0.3%
Case 2	2.333±0.015	2.324±0.030	0.4%
Case 3			
B1	1.41±0.04	1.36±0.04	3.7%
B2	0.90±0.02	0.88±0.04	2.3%
B3	0.87±0.02	0.89±0.04	2.2%
B4	0.86±0.02	0.89±0.04	3.4%
B5	2.21±0.03	2.29±0.05	3.5%

Table 7: Machine Utilizations (Case 1)

	CTL Simulation	DTL Simulation	Error
Block	0.857±0.001	0.857±0.001	0.0%
S_1 Cycle	0.125±0.001	0.125±0.001	0.0%
Down	0.018±0.001	0.018±0.001	0.0%
Cycle	0.125±0.001	0.125±0.001	0.0%
S_2 Down	0.875±0.001	0.875±0.001	0.0%
Starve	0	0	–

Table 8: Machine Utilizations (Case 2)

	CTL Simulation	DTL Simulation	Error
Block	0.252±0.004	0.234±0.006	7.4%
S_1 Cycle	0.187±0.001	0.191±0.001	-2.1%
Down	0.561±0.003	0.575±0.005	-2.4%
Cycle	0.112±0.001	0.115±0.001	-2.6%
S_2 Down	0.591±0.003	0.603±0.003	-2.0%
Starve	0.297±0.002	0.283±0.003	4.9%

Table 9: Machine Utilizations (Case 3)

	CTL Simulation	DTL Simulation	Error
Block	0.186±0.006	0.175±0.007	6.3%
S_1 Cycle	0.752±0.006	0.761±0.008	-1.2%
Down	0.062±0.002	0.064±0.002	-3.1%
Block	0.181±0.006	0.190±0.009	-4.7%
S_2 Cycle	0.591±0.004	0.599±0.006	-1.3%
Down	0.063±0.004	0.061±0.006	3.3%
Starve	0.165±0.003	0.150±0.006	10.0%
Block	0.156±0.006	0.164±0.011	-4.9%
S_3 Cycle	0.591±0.004	0.610±0.001	-3.1%
Down	0.070±0.004	0.069±0.005	1.4%
Starve	0.183±0.004	0.157±0.011	16.6%
Block	0.128±0.005	0.136±0.015	-5.9%
S_4 Cycle	0.591±0.004	0.610±0.001	-3.1%
Down	0.063±0.005	0.068±0.005	-7.4%
Starve	0.218±0.007	0.186±0.017	17.2%
Block	0.098±0.003	0.104±0.008	-5.8%
S_5 Cycle	0.591±0.004	0.610±0.001	-3.1%
Down	0.067±0.006	0.067±0.004	0.0%
Starve	0.244±0.006	0.219±0.009	11.4%
Cycle	0.733±0.005	0.756±0.001	-3.0%
S_6 Down	0.060±0.002	0.061±0.002	-1.6%
Starve	0.207±0.007	0.183±0.002	13.1%

Thus we see that the IPA algorithm already proven for DT lines (Ho, Eyler, and Chien 1983) offers substantial computational savings for gradient computation in CT lines as well.

7 CONCLUSION

For practical purposes, it seems that DT lines can be reasonably represented by CT models. The reason is that one can expect much larger errors in the estimates of input parameters (specially failure and repair rates). The inaccuracies introduced by such errors would overshadow those observed in Tables 6-9.

We have, in fact, experimented with modifying the behavior of the CT model to make it behave "more like a DT line". For instance, in the example at the end of section 5, we can implement a restriction in the model that whenever a machine is starved, it has to wait for a whole unit of product to arrive (i.e. the buffer level must reach unity) before it can start working. Other similar ideas have been explored too. Some of these implementations appear to give estimates that are closer to the PMs of the DT line, but at the cost of several complications. First, we realize that all such ideas introduce more types of events, as well as the consideration of a discrete quantity (a unit product), into the CT models. The resulting model will be more like a simulation model of a DT line which implies more events have to be simulated. This is contrary to one of our aims of using the CT model. Second, the translation of PMs back to the DT line becomes more complicated as well. Third, in our ongoing research, we have successfully implemented IPA algorithms (for gradient estimation) on the basic CT line. With the more complex models, implementation of any type of PA algorithms will be much less obvious. Fourth, the CT model that we analyze corresponds to the way an actual CT line would physically operate. So our present and ongoing work (e.g. gradient estimation) can be used by people designing such lines as well. This would not be the case if we modified the CT model with special rules. Finally, the CT model that we described in this paper provides a *simple* and *intuitive* way to translate the input parameters and PMs from/to a DT line. We feel the ease and simplicity of translation, implementation, and speed of the simpler simulation, outweigh the added accuracy that might be achieved.

Our study also suggests the possibility of applying optimization algorithms to CT models as an approximation to the optimization of DT lines, for example, for the optimal design of buffer sizes and cycle times. It is conceivable that one could use a gradient estimation algorithm, along with the "single run" optimization methods as in Suri and Leung (1987) on a CT model, and then with a final translation algorithm obtain fast estimates of near-optimal parameter settings for DT lines. Our ongoing research is exploring these ideas. We hope our study will provide the foundation for such research, and stimulate additional work on this topic.

Table 10: Gradient Estimates (Cases 1 and 2)

	CFD + formula	IPA
Case 1		
dTP/dC1	2.292×10^{-5}	$(2.262\pm0.327)\times10^{-5}$
dTP/dC2	1.556×10^{-2}	$(1.554\pm0.013)\times10^{-2}$
Case 2		
dTP/dC1	1.817×10^{-2}	$(1.821\pm0.016)\times10^{-2}$
dTP/dC2	3.497×10^{-3}	$(3.498\pm0.044)\times10^{-3}$

Table 11: Input parameters for Case 4

M/c S_i	Flow Rate C_i	Mean Operating Volume to Failure w_i	Repair Rate r_i	Buffer Size $B(i)$	Isolated Throughput TP_i
1	2.857	85.714	0.250	100	2.521
2	4.0	56.0	0.154	100	2.732
3	3.333	116.667	0.100	150	2.593
4	3.125	125.0	0.118	250	2.577
5	3.333	120.0	0.083	250	2.500
6	3.226	45.161	0.286		2.581

Table 12: Gradient Estimates (Case 4)

Case 4	IPA	CFD+Simulation
dTP/dC1	$(1.427\pm0.028)\times10^{-1}$	$(1.434\pm0.024)\times10^{-1}$
dTP/dC2	$(4.513\pm0.052)\times10^{-2}$	$(4.518\pm0.048)\times10^{-2}$
dTP/dC3	$(1.029\pm0.009)\times10^{-1}$	$(1.029\pm0.010)\times10^{-1}$
dTP/dC4	$(9.143\pm0.145)\times10^{-2}$	$(9.176\pm0.123)\times10^{-2}$
dTP/dC5	$(8.148\pm0.234)\times10^{-2}$	$(8.193\pm0.198)\times10^{-2}$
dTP/dC6	$(7.849\pm1.143)\times10^{-3}$	$(7.883\pm1.230)\times10^{-3}$
Simulation time for computing entire gradient vector: IPA: 5090 seconds CFD: 33820 seconds CFD/IPA = 6.64		

ACKNOWLEDGMENTS

This work was partly supported by a grant from Ford Motor Co.

APPENDIX : ALGORITHM FOR SIMULATION AND COMPUTATION OF MULTIPLE GRADIENTS OF A CT LINE

The definitions for the following variables are described in the text: M, t, Δt, e^*, q, Q, C_i, v_i, α_i, $B(j)$, W_i, U_i, and x_j, for $i = 1, \ldots, M$, and $j = 1, \ldots, M-1$, where the argument t is dropped for notational convenience. The following notation is used in the CT line simulation algorithm.

INF: A very large constant.

EM[i]: Next possible event at S_i.

EB[j]: Next possible event at B_j, and *EB[M]* denotes the possible event $\mathcal{T}_f$ if $v_M \neq 0$.
TM[i]: Time to the occurrence of *EM[i]*.
TB[i]: Time to the occurrence of *EB[i]*, and *TB[M]* is the time to the occurrence of *EB[M]*.
K_M, K_B: Constants.
K^*: Index of triggering event.
L(j): Accumulator for average level of B_j.
TSum[k]: $M \times 1$ array of accumulators.
QSum[i][k], USum[i][k]: $M \times M$ array of accumulators.

α, C, v, W, U, x, and L are the corresponding column vectors. The simulation algorithm is shown in 5 steps and three procedures. The IPA algorithm is added to the simulation algorithm in the steps marked "IPA".

0. SYSTEM INITIALIZATION
$t = 0$; $\Delta t = 0$; $\alpha = O$; $x = 0$; $L = 0$; $q = Q$; $x_0 = INF$;
$B(M) = INF$; $v_1 = C_1$; Generate W; $EB[M] = \mathcal{T}_f$;
for $i = 2$ to M do $v_i = min\ (v_{i-1}, C_i)$;
0-IPA. INITIALIZATION
for $k = 1$ to M do {
$TSum[k] = 0$;
for $i = 1$ to M do
{ $QSum[i][k] = 0$; $USum[i][k] = 0$; } }
1. NEXT EVENT AT MACHINES AND BUFFERS
for $j = 1$ to $M-1$ do {
case
$v_j > v_{j+1}$: { $TB[j] = (B(j) - x_j)/(v_j - v_{j+1})$;
$EB[j] = \mathcal{BF}_j$; }
$v_j < v_{j+1}$: { $TB[j] = x_j/(v_{j+1} - v_j)$;
$EB[j] = \mathcal{BE}_j$; }
$v_j = v_{j+1}$: $TB[j] = INF$;
end case }
for $i = 1$ to M do {
if $(\alpha_i \neq D)$ then
if $(v_i = 0)$ then $TM[i] = INF$;
else { $EM[i] = \mathcal{F}_i$; $TM[i] = W_i/v_i$; }
else
{ $EM[i] = \mathcal{R}_i$; $TM[i] = U_i$; } }
if $(v_M \neq 0)$ then $TB[M] = q/v_M$;
else $TB[M] = INF$;
2. NEXT EVENT IN CTL
$K_M = argmin\ (TM[i])$; $i = 1, 2, ..., M$.
$K_B = argmin\ (TB[i])$; $i = 1, 2, ..., M$.
if $(TM[K_M] > TB[K_B])$ then
{ $K^* = K_B$; $\Delta t = TB[K]$; $e^* = EB[K]$; }
else { $K^* = K_M$; $\Delta t = TM[K]$; $e^* = EM[K]$; }
3A-IPA. PERTURBATION GENERATION
If ($e^* = F_{K^*}$) then
for $k = 1$ to M do {
$temp[k] = -\{ \tau \cdot 1(K^*, k) + QSum[K^*][k] \}/v[K^*]$;
$USum[K^*][k] = 0$; }
If ($e^* = R_{K^*}$) then
for $k = 1$ to M do $temp[k] = USum[K^*][k]$;
If ($e^* = BF_{K^*}$) then
for $k = 1$ to M do
$temp[k] = -\{ \tau \cdot 1(K^*, k) - 1(K^*+1, k) \} +$
$QSum[K^*][k] - QSum[K^*+1][k] \}/(v[K^*] - v[K^*+1])$;
If ($e^* = BE_{K^*}$) then
for $k = 1$ to M do
$temp[k] = -\{ \tau \cdot 1(K^*+1, k) - 1(K^*, k) \} +$
$QSum[K^*+1][k] - QSum[K^*][k] \}/(v[K^*+1] - v[K^*])$;
If ($e^* = T_f$) then
for $k = 1$ to M do
$temp[k] = -\{ \tau \cdot 1(M,k) + QSum[M][k] \}/v[M]$;
where $1(m, k) = \begin{cases} 1, & \text{if } v[m] = C_k, \\ 0, & \text{otherwise.} \end{cases}$
3B-IPA. PERTURBATION UPDATE
for $k = 1$ to M do {
$TSum[k] = TSum[k] + temp[k]$;
for $i = 1$ to M do
if (S_i is failed) then
$USum[i][k] = USum[i][k] - temp[k]$;
else
$QSum[i][k] = QSum[i][k] + \tau \cdot 1(i,k)$
$+ v[i] \cdot temp[k]$; }
3. SYSTEM UPDATE
$t = t + \Delta t$; $q = q - v_M \cdot \Delta t$;
$w = w - v \cdot \Delta t$; $u = u - \tau \cdot 1$;
for $j = 1$ to $M - 1$ do {
$temp = x_j$; $x_j = x_j + (v_j - v_{j+1}) \cdot \Delta t$;
$L(j) = L(j) + (temp + x_j) \cdot \Delta t/2$; }
case
$e^* = \mathcal{F}_{K^*}$: { $\alpha_{K^*} = D$; $v_{K^*} = 0$; $U_i = sample(1/r_i)$;
Update_Down_mcs (K^*);
Update_Up_mcs ($K^* - 1$); }
$e^* = \mathcal{R}_{K^*}$: { Set_Flow_Rate (K^*);
$W_i = sample(w_i)$;
Update_Down_mcs (K^*);
Update_Up_mcs ($K^* - 1$); }
$e^* = \mathcal{BF}_{K^*}$: Update_Up_mcs (K^*);
$e^* = \mathcal{BE}_{K^*}$: Update_Down_mcs (K^*);
end case
4. TERMINATION TEST
if ($e^* = \mathcal{T}_f$) then go to OUTPUT;
else go to NEXT EVENT AT MACHINES AND BUFFERS;
5. PMs OUTPUT
Throughput = Q/t ;
Average buffer levels = L/t ;
5-IPA. GRADIENT OUTPUT
for $k = 1$ to M do { $G_k = -Q \cdot TSum[k]/t^2$; }

procedure Set_Flow_Rate (n)
case
$x_{n-1} = 0$ and $x_n = B(n)$: $v_n = min\ (v_{n-1}, C_n, v_{n+1})$;
$x_{n-1} = 0$ and $x_n < B(n)$: $v_n = min\ (v_{n-1}, C_n)$;
$x_{n-1} > 0$ and $x_n = B(n)$: $v_n = min\ (C_n, v_{n+1})$;
otherwise: { $v_n = C_n$; $\alpha_i = O$; }
end case;

```
  if (v_n = v_n-1) then α_n = S;
  if (v_n = v_n+1) then α_n = B;
procedure Update_Down_mcs (n)
  while ((x_n is empty) and (n ≤ M − 1)) do
    { n = n + 1; Set_Flow_Rate (n); }
procedure Update_Up_mcs (n)
  while ((x_n is full) and (n ≥ 1)) do
    { Set_Flow_Rate (n); n = n − 1; }
```

REFERENCES

Altiok, T., and S. Stidham, Jr. 1982. A Note on Transfer Lines with Unreliable Machines, Random Processing Times, and Finite Buffers. *IIE Transactions*. Vol. 14, No. 2, 125-127.

Alvarez, R., Y. Dallery, and R. David. 1991. An Experimental Study of the Continuous Flow Model of Transfer Lines with Unreliable Machines and Finite Buffers. In IMACS-IFAC SYMPOSIUM Modeling and Control of Technological Systems 1991.

Choong, Y. F., and S. B. Gershwin. 1987. A Decomposition Method for The Approximate Evaluation of Capacitated Transfer Lines with Unreliable Machines and Random Processing Times. *IIE Transactions*. Vol. 19, No. 2, 150-159.

Dallery, Y., R. David, and X.-L. Xie. 1988. An Efficient Algorithm for Analysis of Transfer Lines with Unreliable Machines and Finite Buffers. *IIE Transactions*. Vol. 20, No. 3, 280-283.

Dallery, Y., R. David, and X.-L. Xie. 1989. Approximate Analysis of Transfer Lines with Unreliable Machines and Finite Buffers. *IEEE Transactions on Automatic Control*. Vol. 34, No. 9, 943-953.

Gershwin, S. B., and I. C. Schick. 1980. Continuous Model of an Unreliable Two-Stage Material Flow System with a Finite Interstage Buffer. MIT Technical Report. LIDS-R-1039.

Gershwin, S. B., and I. C. Schick. 1983. Modeling and Analysis of Three-Stage Transfer Lines with Unreliable Machines and Finite Buffers. *Operations Research*. Vol. 31, No. 2, 354-380.

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer Academic Publishers.

Glynn, P. W. 1986. Optimization of Stochastic Systems. In *Proceedings of the 1986 Winter Simulation Conference*, eds. J. Wilson, J. Henriksen, and S. Roberts, 52-59.

Glynn, P. W. 1989. A GSMP Formalism for Discrete Event Systems. *Proceedings of the IEEE*. Vol. 77, No. 1, 14-23.

Ho, Y. C., M. A. Eyler, and T. T. Chien. 1983. A New Approach to Determine Parameter Sensitivities of Transfer Lines. *Management Science*. Vol. 29, No. 6. 700-714.

Koster, R. D., and J. Wijngaard. 1989. Continuous vs. Discrete Models for Production Lines with Blocking. *Queueing Networks with Blocking*. H. G. Perros and T. Altiok eds. North-Holland.

Suri, R. 1989. Perturbation Analysis: The State of the Art and Research Issues Explained via the GI/G/1 Queue. *Proceedings of the IEEE*. Vol. 77, No. 1, 114-137.

Suri, R., and B.-R. Fu. 1991a. On Using Continuous Lines to Model Discrete Production Lines: Part I – Representation. University of Wisconsin-Madison. Technical Report.

Suri, R., and B.-R. Fu. 1991b. On Using Continuous Lines to Model Discrete Production Lines: Part II – Cycle Time Gradient Estimation. University of Wisconsin-Madison. Technical Report.

Suri, R. and Y. T. Leung. 1987. Single Run Optimization of a SIMAN Model for Closed Loop Flexible Assembly Systems. In *Proceedings of the 1987 Winter Simulation Conference*, eds. A. Thesen, H. Grant, W.D. Kelton, 738-748.

Suri, R., J. Sanders, and M. Kamath. 1992. Performance Evaluation of Production Networks. Chapter in *Handbook in Operations Research, Vol. 4*, S. C. Graves, A. H. G. Rinnooy Kan and P. H. Zipkin eds. Elsevier (to appear).

Wei, K. C., Q. Q. Tsao, and N. C. Otto. 1989. Determining Buffer Size Requirements Using Stochastic Approximation Methods. Technical Report. SR-89-73. Ford Research.

AUTHOR BIOGRAPHIES

RAJAN SURI is Professor of Industrial Engineering at the University of Wisconsin - Madison, and one of the faculty responsible for the Manufacturing Systems Engineering Program. He received his Bachelor's degree from Cambridge University (England), and his M.S. and Ph.D. degrees from Harvard University. He has been instrumental in extending the theories of queueing networks and perturbation analysis for manufacturing applications and is the author of many publications. He is Editor-in-Chief of the *Journal of Manufacturing Systems*, Associated Editor of the *International Journal of Flexible Manufacturing Systems*, and Area Editor of the *Journal of Discrete Event Dynamic Systems*. Dr. Suri has consulted for leading firms including 3M, Alcoa, AT&T, DEC, FIAT, Ford, Hewlett Packard, IBM, Pratt & Whitney, and Siemens. He is also a principal of Network Dynamics, Inc., a firm specializing in software for manufacturing systems. In 1981 Dr. Suri received the Eckman Award for outstanding contributions from the American Automatic Control Council.

BOR-RUEY FU is a Ph.D. candidate in the Department of Industrial Engineering at the University of Wisconsin - Madison. He received a B.S. in Industrial Engineering from National Tsing Hua University, Taiwan in 1983. His current research interests are in design and analysis of manufacturing systems, stochastic optimization, and discrete-event simulation. He has worked on projects with several Wisconsin firms. He is a member of ORSA, TIMS, IIE, and SME.