

Copyright © 1990 IEEE. Reprinted from *IEEE Transactions on Robotics and Automation*.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Columbia Business School's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

Efficient Computation of Coordinating Controls in Hierarchical Structures for Failure-Prone Multi-Cell Flexible Assembly Systems

RAMAKRISHNA AKELLA, MEMBER, IEEE, BRUCE H. KROGH, MEMBER, IEEE, AND MEDINI R. SINGH

Abstract—This paper concerns production allocation in multicell manufacturing systems. A model is developed for a hierarchical control scheme, where each cell consists of several failure-prone machines for which the time-scale of the machine state transitions is comparable to the processing times. The production objective is to track a nonstationary demand as closely as possible when the demand is near or exceeds the capacity of the system. The contribution of this paper is threefold. First, a series of approximations are proposed to obtain a model that is realistic while admitting a tractable solution. Second, to solve the resulting stochastic control problem, we derive a general result on the second-order finite-time (transient) statistics of a continuous-time Markov chain. Finally, simulation results are presented to illustrate the proposed model and control methodology. These results are compared with a myopic linear programming approach.

I. INTRODUCTION

RECENTLY, there has been a great deal of interest in the design, planning, real-time dispatch, and control of multicell systems for electronic and computer assembly. Results have been mainly limited, however, to real-time dispatch for single cell, high-volume systems with failure-prone machines and constant demand rates [1], [2], [4], [8], [9]. As pointed out in the seminal paper [9], these results are applicable only to situations when machine state changes occur on a much slower time-scale than the processing times. In contrast, here we develop a real-time dispatch procedure for system with a large number of machines (which can be located in a number of cells). In such systems, the machine state changes occur on a time-scale comparable to the processing times. The objective for the single/multiple-cell system is to track a nonstationary demand as closely as possible when the demand is near or exceeds the capacity of the system.

The current work has been motivated by an electronic

Manuscript received June 1988; revised May 1990. Approved for publication by A. Desrochers. This work was supported by an IBM Faculty Development Award in Manufacturing, an IBM Charlotte Contract, an NSF Grant under the Manufacturing Program of the Engineering Design Research Center, NSF Grant DMC-8451493, North American Phillips Corporation, Xerox Corporation and an IBM Graduate Fellowship for Manufacturing Research, T. J. Watson Research Center, Yorktown Heights, NY.

R. Akella is with the Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213.

B. H. Krogh is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213.

M. R. Singh is with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109.

IEEE Log Number 9038619.

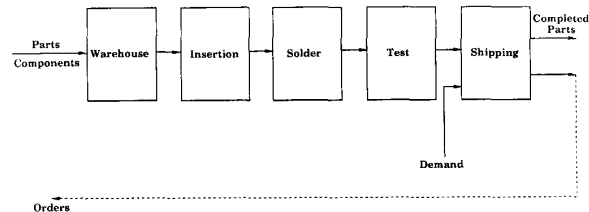


Fig. 1. Cardlines for electronic assembly.

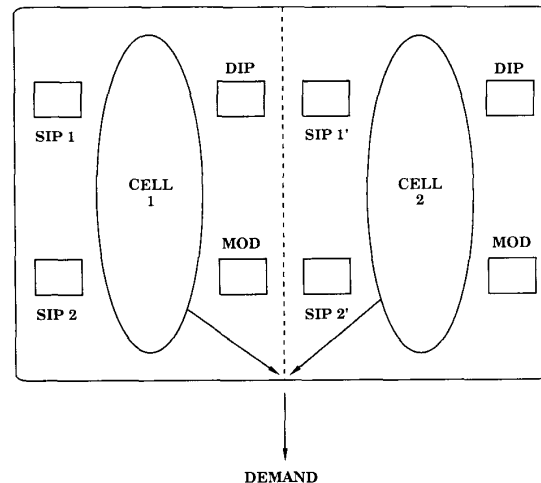


Fig. 2. Parallel lines/cells.

assembly facility. A typical card-line for electronic assembly, which is shown in Fig. 1, consists of several stages; of these, we focus on the insertion stage. A typical insertion stage is shown in Fig. 2, where components of different types are inserted at various machines. The line consists of multiple cells in parallel, with virtually no communication among them. A coordinator is responsible for periodic communication and control of these cells. Machine availability is the dominant uncertainty in each of the cells due to machine failures and repairs. A coordinator allocates production rates to be implemented during a coordination period, using the machine state, demand, and inventory positions that are available at the beginning of each period. The next demand arrives at the beginning of the next coordination period. The

production rates specified by the coordinator are implemented by the local controllers as a function of the instantaneous machine state of the local cell. Cell controllers do not communicate with the coordinator or the other cells except at the coordination epochs.

A key concept in the present work is that of the aggregation of the instantaneous capacity constraints. This results in the dispatch policy being a function of the random amount of time to be spent in each state. The production quantity can then be expressed in terms of the yet-to-be-determined production rates in a given state and the random times spent in each of these states. A key parameter in the proposed aggregation scheme is the second-order finite-time (transient) statistics of the continuous-time Markov chain governing state transitions. We obtain a closed-form solution for this quantity, which in itself is a result of significance and interest.

A major hurdle in the control of systems whose capacity is subject to random fluctuations governed by a continuous-time, discrete-state Markov chain with extremely large state space is the simultaneous integration of a large set of strongly coupled Riccati-type differential equations [14]. For the real-time control of manufacturing systems, algorithms that are computationally simple and less demanding, albeit suboptimal, can prove very useful. The current work is an attempt in this direction. Through a series of approximations, we obtain a model that is computationally tractable and involves little on-line computation.

In a recent series of papers by Shanthikumar and Yao [10], [11], [13], [16], [17], the issues of buffer space, machine, and production capacity allocation across different cells have been extensively treated. Their approach is based on queuing models, followed by optimization. Some of the results in that work correspond to production allocations across cells with stationary demand. However, their approach does not include dynamic capacity constraints. In contrast, we consider here the case of nonstationary demand in the presence of machine failures in a multicell situation, where simultaneous dynamic capacity constraints are imposed. Our model of the real-time control of each cell in the system follows previous formulations of the single-cell problem [2], [4], [8].

The paper is organized as follows. In the next section, we describe and formulate the multicell coordination problem. The derived control policy requires the computation of finite-time statistics. Section III summarizes the results related to the computation of finite-time (transient) statistics. A brief summary of computational steps to implement the coordinating control is given in Section IV. For purposes of comparison, we provide a myopic linear programming formulation of the problem in Section V. A study of the effectiveness of the proposed algorithm vis-a-vis the myopic LP solution is done using discrete-event simulation of a two-cell system. The results are summarized in Section VI.

II. CONTROL OF MULTICELL SYSTEMS WITH FAILURE PRONE MACHINES

We consider a manufacturing system producing a number of products. The demand for each product is known for the

future N periods, and the goal is to try to meet this demand. The manufacturing system consists of a number of parallel cells, where each has a combination of different machines. The grouping of machines into cells is influenced by several design considerations, ranging from layout to capacities of machines, material handling systems, etc. Each cell can produce a number of different products (similarly a product may be produced in a number of alternative cells), but there is no flow of material or information between the cells. During a given period of operation, a major source of uncertainty is capacity changes due to machine failures and repairs. We consider the situation when the system is heavily loaded, that is, the demand for parts is nearly equal to the total system capacity. Thus, an efficient allocation of production resources is critical, and machine failures must be anticipated to obtain a good performance.

Associated with each product, there are inventory holding and backlogging costs that may change from period to period. Machines are flexible in the sense that they can process a group of products with virtually no setups. This gives the flexibility of producing any group of products on a set of machines simultaneously but not necessarily at the same production rate. These machines may physically be located in different cells with no information exchange. The real-time control problem can be stated the following way: Given the machine states, the current inventory status, and the future requirements, what should be produced on each machine such that the sum of inventory holding and backlogging costs for all the products is minimized over the N -period time horizon? The availability of the raw components is ensured by a higher-level model that deals with the medium-term planning.

In a manufacturing system with a number of cells (each consisting of a large number of machines), the total number of possible machine states grows exponentially with the number of machines. Designing a centralized controller that can respond to every change in machine states by recomputing the control policy and changing the dispatch quantity for all the machines will be computationally prohibitive. For the real-time implementation, the feedback of information as well as the computation of dispatch quantities should be instantaneous. We adopt a two-level hierarchical model (shown in Fig. 3), where each cell is controlled by a cell controller, which receives information about the machine states and specifies the cell production rate on a second-by-second basis. The coordination between the cells is achieved periodically by a cell coordinator that receives the feedback from the inventory as well as the cell machine states at time intervals of period T , which is of the order of a few hours. Based on this information and also on the cost parameters and the future requirements, the cell coordinator specifies production rates for each cell corresponding to every possible cell state. The cell controllers use these values until the next coordination epoch, when new updates are again downloaded. Clearly, the computation is still centralized but it is done on a periodic basis, which gives the cell coordinator enough time to carry out the dynamic program.

This control architecture has been adopted due to the

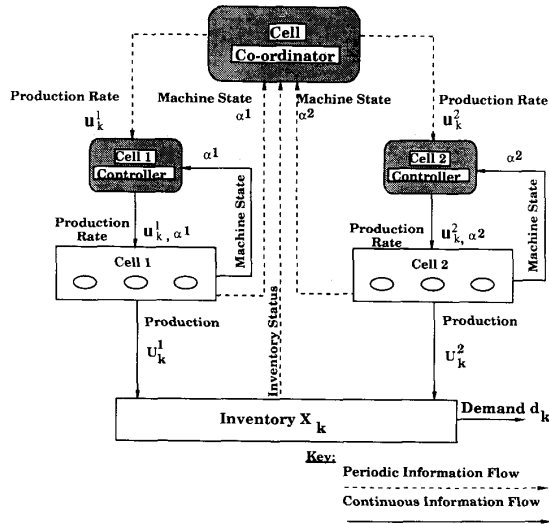


Fig. 3. Control architecture.

limitations on the computational speed of the processor and the capacity of the communication cables. In addition, the communication network and the host tend to go down several times a year, interrupting the communication between the cell and the coordinator for long time. The hierarchical architecture described above has the advantage of not disrupting the production under such circumstances. A centralized controller with more frequent updates can be used to advantage as future technology makes communication and computation faster and more reliable.

Notation: We define the following variables and vectors that will be used later in the paper:

p	number of products
n	number of cells
T	cell coordination interval
c	superscript for cell ($c = 1, 2, \dots, n$)
i	index for product ($i = 1, 2, \dots, p$)
k	index for period ($k = 1, 2, \dots, N$)
J^c	number of types of machines in cell c
$y^c(t)$	continuous-time Markov chain representing state of cell c at time t
s^c	number of machine-states for cell c
j	index for machine type ($j = 1, 2, \dots, J^c$)
α^c	state of cell c ($\alpha^c = 1, 2, \dots, S^c$)
$d_k(i)$	demand for product i during k th period
$x_k(i)$	stock available for product i at the beginning of the k th period
$u_{k, \alpha^c}^c(i)$	production rate for product i in cell c during period k if the cell is in state α^c ; Decision variable for cell coordinator (number of units/min)
$U_k^c(i)$	random production quantity of product i in cell c during period k as a result of the state changes
$a_{\alpha^c}^c(j)$	number of type j machines up in cell c if the cell is in state α^c .
$\tau_{k, \alpha^c}^c(T)$	random amount of time cell c spends in state α^c during period k of length T .

$D_{(j,i)}^c$ amount of time a type j machine will take to process a unit of product i ; $i = 1, 2, \dots, p$; $j = 1, 2, \dots, J^c$

I identity matrix of dimension $p \times p$:

$$x_k = \begin{bmatrix} x_k(1) \\ \vdots \\ x_k(i) \\ \vdots \\ x_k(p) \end{bmatrix}; d_k = \begin{bmatrix} d_k(1) \\ \vdots \\ d_k(i) \\ \vdots \\ d_k(p) \end{bmatrix}; U_k^c = \begin{bmatrix} U_k^c(1) \\ \vdots \\ U_k^c(i) \\ \vdots \\ U_k^c(p) \end{bmatrix};$$

$$a_{\alpha^c}^c = \begin{bmatrix} a_{\alpha^c}^c(1) \\ \vdots \\ a_{\alpha^c}^c(i) \\ \vdots \\ a_{\alpha^c}^c(p) \end{bmatrix}; a^c = \begin{bmatrix} a_1^c \\ \vdots \\ a_{\alpha^c}^c \\ \vdots \\ a_{S^c}^c \end{bmatrix}; a = \begin{bmatrix} a^1 \\ \vdots \\ a^c \\ \vdots \\ a^n \end{bmatrix};$$

$$u_{k, \alpha^c}^c = \begin{bmatrix} u_{k, \alpha^c}^c(1) \\ \vdots \\ u_{k, \alpha^c}^c(i) \\ \vdots \\ u_{k, \alpha^c}^c(p) \end{bmatrix}; u_k^c = \begin{bmatrix} u_{k, 1}^c \\ \vdots \\ u_{k, \alpha^c}^c \\ \vdots \\ u_{k, S^c}^c \end{bmatrix}; u_k = \begin{bmatrix} u_k^1 \\ \vdots \\ u_k^c \\ \vdots \\ u_k^n \end{bmatrix};$$

$$D^c = \begin{bmatrix} D^c(1,1) & \cdots & D^c(1,p) \\ \vdots & D^c(j,i) & \vdots \\ D^c(J^c,1) & \cdots & D^c(J^c,p) \end{bmatrix};$$

$$\underline{D}^c = \text{diag} [\underline{D}^c \underline{D}^c \cdots \underline{D}^c \{S^c \text{ times}\}];$$

$$\underline{D} = \text{diag} [\underline{D}^1 \cdots \underline{D}^c \cdots \underline{D}^n]$$

$$\tau_k^c = [I\tau_{k,1}^c \cdots I\tau_{k,\alpha^c}^c \cdots I\tau_{k,S^c}^c];$$

$$B_k = [\tau_k^1 \cdots \tau_k^c \cdots \tau_k^n].$$

A. Problem Formulation

Consider a cell c with a group of machines, where each can be in either functional or breakdown state. The transition between these two states is modeled as a continuous-time Markov chain. We assume that the repair starts as soon as a machine fails and that the failure and the repair of each machine is independent of the state of the other machines. Let the total cell state α^c have the information about the number of functional machines of each type. The transition between the cell states can then be represented by an equivalent finite-state continuous-time Markov chain $y^c(t)$. The transition rates of this Markov chain will be a function of the mean time between failure (MTBF) and mean time to repair (MTTR) of individual machines.

A quantity of fundamental importance in deciding the

production dynamics is $\tau_{k,\alpha^c}^c(T)$, which is the random amount of time cell c spends in state α^c during period k and is expressed as

$$\tau_{k,\alpha^c}^c(T) = \int_{(k-1)T}^{kT} y_{\alpha^c}^c(t) dt$$

where

$$y_{\alpha^c}^c(t) = \begin{cases} 1 & \text{if } y^c(t) = \alpha^c, \text{ i.e., at time } t, \text{ the cell } c \text{ is} \\ & \text{in state } \alpha^c \\ 0 & \text{otherwise.} \end{cases}$$

Suppose for period k the cell coordinator specifies the vector of production rates u_{k,α^c}^c for all cell states α^c that are implemented by the cell controller. The actual production of cell c during period k will then be

$$U_k^c = \sum_{\alpha^c=1}^{S^c} \tau_{k,\alpha^c}^c u_{k,\alpha^c}^c. \quad (1)$$

We note that U_k^c is a random vector because the amount of time cell c spends in each state is random. Letting u_k be the vector of production rates for all the cells and defining B_k as matrix of random times $\tau_{k,\alpha^c}^c(T)$ as defined above, (1) gives

$$U_k = \sum_{c=1}^n U_k^c = B_k u_k$$

where U_k is the vector of total production by all cells. The inventory balance equation for the multicell system can be written as

$$x_{k+1} = x_k + B_k u_k - d_k \quad (2)$$

where negative inventory values represent the backlog that must be met by future production. The specified production rates u_{k,α^c}^c must be within the production capacity of the functional machines, i.e.

$$D^c u_{k,\alpha^c}^c \leq a_{\alpha^c}^c \forall \alpha^c$$

where D^c is the matrix of processing times, and the components of the vector $a_{\alpha^c}^c$ represent the number of functional machines of each type. Stacking the above equations for $\alpha^c = 1, 2, \dots, S^c$ gives

$$D^c u_k^c \leq a^c. \quad (3)$$

The above equations and constraints holds for each cell of the manufacturing system. It is convenient to rewrite (3) for $c = 1, 2, \dots, n$, in a more compact form

$$D u_k \leq a \quad (4)$$

where u_k is the vector of production rates specified by the cell coordinator, which tells each cell, for every possible state, what to produce, on which type machines, and at what rate, and vector a represents the number of machines of each type available in each cell for every state. We shall also require the production rates u_k to be nonnegative. Unfortunately, incorporating this hard constraint explicitly in the present formulation is difficult. We propose to impose this constraint heuristically by setting any negative production rate to zero.

We now turn to the task of computing the production rates u_k , $k = 1, 2, \dots, N$ so that a given cost function

$$E \left\{ \sum_{k=1}^N g_k(x_k) \right\}$$

is minimized subject to the inventory balance equations (2) and capacity constraints (4), where $g_k(x_k)$ can be a piecewise linear, quadratic, or any general convex function, depending upon how inventory costs are incurred.

B. Problem Solution

An exact solution of the above problem for a general cost function is intractable. To obtain a computationally feasible approximate solution to the above problem, we make the following approximation, following [1], where an efficient solution for the single-cell problem is described. We approximate the cost function by the quadratic form

$$g(x_k) \approx x_k^T Q_k x_k - C_k x_k$$

where elements of the diagonal matrix Q_k and the vector C_k are chosen to fit the corresponding inventory holding and backlogging cost curves. The second term in the above approximation appears to mimic the asymmetry in cost functions corresponding to the positive and the negative inventories.

Our second approximation converts the constrained optimization problem into an unconstrained optimization problem. In particular, the capacity restrictions (4) are enforced by adding a cost

$$(D u_k - a)^T R_k (D u_k - a) + W_k (D u_k - a)$$

in the objective function, where the elements of matrix R_k ensure that the capacity violations are heavily penalized. Vector W_k can be used to mitigate the penalty for not using the available capacity, although this will rarely happen in the proposed model. The reason is as follows: If the demand is low and the machines are consistently underutilized, then an immediate response to machine failure/repair is not critical; the problem is trivial in this case because the optimal decision for each period is to produce until all demands are satisfied. On the other hand, when the capacity is scarce, a judicious utilization of available machines becomes critical. It is under these circumstances that the proposed control rule proves its value. We also note that if a machine type is consistently underutilized, it can then be excluded from the constraint set (4), or equivalently, the corresponding elements of matrix R_k can be set to zero. In many manufacturing systems, capacity costs are sunk for the short-term decisions, and any underutilized capacity incurs a loss due to numerous fixed costs. It is reasonable, in such circumstances, to include the cost of underutilization in the formulation.

With the above approximations, the cell coordination problem can be written as

$$\min_{u_k} E \left\{ \sum_{k=1}^N x_k^T Q_k x_k - C_k x_k + (D u_k - a)^T \cdot R_k (D u_k - a) + W_k (D u_k - a) \right\} \quad (5)$$

subject to the inventory balance equation (2). The formulation now is easily recognized as the linear-quadratic control problem with jump parameters. Linear systems whose parameters are subject to random fluctuations have been studied

by, among others. Sworder [14] and Wonham [15] for the continuous-time case and by Blair and Sworder [6] and Chizeck *et al.* [7] for the discrete-time case. However, the control problem defined above cannot be addressed effectively in either of the two frameworks. Although the periodic shipment of demand and associated bookkeeping leads to discrete-time inventory balance equations and cost functional similar to the discrete-time jump-linear system in [6] and [7], a major difference is that the capacity may change here many times within a period. This distinction can also be noticed by the fact that the matrix B_k here is stochastic even when the cell state at the beginning of period k is known. Formulating the problem as a continuous-time linear system with randomly jumping parameters, as in [14], would have lead to a control law requiring numerical integration of a coupled set of Ricatti-type equations—a task that is unrealistic for even small problems. In contrast, the proposed formulation yields a tractable solution through the approximation in the following section.

C. Quasi-Steady-State Approximation

In order to obtain a recursive form of solution, we need to ensure that the B_k 's for the successive periods are independent. The following discussion provides a justification for this assumption. As discussed earlier, the elements of B_k 's correspond to the random amount of times the cells spend in different states during the k th period. In general, the fraction of time a cell spends in any state during a coordination period depends on the initial state of the cell. This dependence becomes weaker, however, as the length of coordination period increases. In particular, as the period T increases, the underlying Markov chain leads to stationary probabilities given by

$$\lim_{T \rightarrow \infty} \frac{E\{\tau_{k,\alpha^c}^c(T)\}}{T} = \pi_{\alpha^c}^c$$

where $\pi_{\alpha^c}^c$ is the stationary probability corresponding to state α^c .

Moreover, as T increases, the occupation times become independent, that is

$$\lim_{T \rightarrow \infty} \frac{E\{\tau_{k,\alpha^c}^c(T)\tau_{k,\alpha^c}^c(T)\}}{T^2} = \pi_{\alpha^c}^c \pi_{\alpha^c}^c$$

which means that for large T , not only the computations of B_k 's are trivial, but in addition, there is no correlation between elements of the coefficients matrices for successive periods which is a property we need for efficient computation of controls.

Unfortunately, a long coordination period also means slower updating of the inventory status and cell production rates. Clearly, an efficient computational method at the cost of valuable information and resultant performance degradation is undesirable. To resolve this dilemma, we choose the length of coordination period that is long enough so that the fraction of time spent in (dominant) states reach their stationary values. One should remember, the larger the number of machines of each type in a given cell, the larger the number of transitions and the less time it will take to reach the stationary probabilities. For a given cell structure, we can

arrive at the coordination period that ensures the aforementioned condition. We can then assume that the statistics for the first period depend on the initial state, but those for the subsequent periods are independent of each other. This is equivalent to making the decision for the current period based on the present state while using averages for periods far out in the future. The computations are repeated at the beginning of future periods using the same assumptions. Thus, we propose a moving-horizon open-loop feedback control strategy.

The approximation that the random coefficient matrices are independent for successive periods permits us to apply the results of stochastic dynamic programming [3], [5] which yields production decisions as a linear function of current production status given by

$$u_k = L_k x_k + M_k \quad (6)$$

where

$$\left. \begin{aligned} \bar{B}_k &= E\{B_k\} \\ F_k &= [E\{B_k^T K_{k+1} B_k\} + D^T R_k D]^{-1} \\ L_k &= -F_k \bar{B}_k^T K_{k+1} \\ M_k &= F_k [D^T R_k a + \bar{B}_k K_{k+1} d_k \\ &\quad - (P_{k+1} \bar{B}_k)^T - \frac{1}{2} (W_k D)^T] \\ K_k &= Q_k - K_{k+1} \bar{B}_k F_k \bar{B}_k^T K_{k+1} + K_{k+1} \\ P_k &= P_{k+1} M_k^T \bar{B}_k^T K_{k+1} - d_k^T K_{k+1} - \frac{1}{2} C_k \\ P_N &= -\frac{1}{2} C_N; K_N = Q_N. \end{aligned} \right\} \quad (7)$$

Computation of $E\{B_k\}$ and $E\{B_k^T K_{k+1} B_k\}$ entails calculation of terms like $E\{\tau_{k,\alpha^c}^c(T)\}$ and $E\{\tau_{k,\alpha^c}^c(T) \tau_{k,\alpha^c}^c(T)\}$. Under stationarity assumption, these terms can be readily computed for stages $k = 2, 3, \dots, N$ as

$$\begin{aligned} E\{\tau_{k,\alpha^c}^c(T)\} &\cong T \pi_{\alpha^c}^c \\ E\{\tau_{k,\alpha^c}^c(T) \tau_{k,\alpha^c}^c(T)\} &\cong T^c \pi_{\alpha^c}^c \pi_{\alpha^c}^c. \end{aligned}$$

However, we do not want to use the limiting probability distribution for the first period when the transient behavior of the Markov chain is important. Rather, we use the finite-time statistics $E\{\tau_{k,\alpha^c}^c(T) | y^c(0)\}$ and $E\{\tau_{k,\alpha^c}^c(T) \cdot \tau_{k,\alpha^c}^c(T) | y^c(0)\}$, which take advantage of the available information about the current cell states.

Note that under the limiting assumption, all matrices in (7) can be calculated recursively (starting from the last period) up to the second period without any feedback about the inventory or the cell states. It is the coefficient matrices for the first period that need information about the current cell states. The present inventory status is needed to compute the production rates for the first period using the linear decision rule (6). In the next coordination period, we repeat this whole exercise (a rolling horizon approach), and the production rates for the next period will be based on the cell states at the beginning of *that* period. The proposed solution technique, in effect, uses the current state to take the production decision for the immediate future, whereas the steady-state distribution is used to assess the effect of future demands and costs. We emphasize that the computation of the

right production rates is contingent on an accurate evaluation of finite-time statistics $E\{\tau_{k,\alpha^c}^c(T) | y^c(0)\}$ and $E\{\tau_{k,\alpha^c}^c(T) \tau_{k,\alpha^c}^c(T) | y^c(0)\}$. The next section addresses this aspect of the problem.

III. COMPUTATION OF FINITE-TIME STATISTICS

For notational simplicity, explicit reference to cell c or period k is suppressed in the following development. As a matter of fact, since states $\alpha^c, \alpha'^c, y^c(0) \in \{1, 2, \dots, S^c\}$, we will refer to cell states simply by letters j, j' etc., indicating integer values. Thus, we shall write $E\{\tau_{k,\alpha^c}^c(T) \tau_{k,\alpha^c}^c(T) | y^c(0)\}$ as $E\{\tau_\alpha(T) \tau_{\alpha'}(T) | y(0)\}$ or simply as $E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\}$.

Theorem 1: Let $y(t)$ be a continuous-time Markov chain with generator Q , and let $\tau_j(T)$ and $\tau_{j'}(T)$ be the occupation times in states j and j' , respectively. Then, the joint statistics of occupation times conditioned on initial state $y(0)$ is given by

$$E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\} = \left[\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \sum_{\forall i,l,m | i+l+m=N} (Q^i \delta_j Q^l \delta_{j'} Q^m + Q^i \delta_{j'} Q^l \delta_j Q^m) \mathbf{1} \right]_k$$

where $\mathbf{1}$ is a vector of 1, $\delta_j = \begin{bmatrix} \delta_{j1} & & & \\ & \delta_{j2} & & \\ & & \ddots & \\ & & & \delta_{jS^c} \end{bmatrix}$, and

δ_{jk} is the Kronecker delta function.

Proof: See the Appendix.

The convergence of the infinite series is guaranteed because eigenvalues of the generator matrix Q are finite. Irrespective of the value of T , the factorial in the denominator will ultimately dominate, making the successive terms in the series diminish. To evaluate the computational burden, we first express the series solution in a computationally more amenable form, namely

$$E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\} = \sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!}$$

$$\sum_{\forall i,l,m | i+l+m=N} \left([Q^i]_{kj} [Q^l]_{j'j'} \sum_{\nu} [Q^m]_{j\nu} + [Q^i]_{kj'} [Q^l]_{j'j} \sum_{\nu} [Q^m]_{j\nu} \right)$$

The spectral representation of the generator $Q = \Phi \Lambda \Psi$ can be used to compute the matrix exponents such that $[Q^i]_{kj} = \sum_r \phi_{kr} \lambda_r^i \psi_{rj}$, where $\Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_{S^c}]$ is the eigenvalue matrix, and Φ and Ψ are the left and right eigenvectors, respectively.

The computational complexity in evaluating the series sum can be judged by the fact that corresponding to each N , there will be exactly $(N+1)(N+2)$ terms in the inner summation, where each requires $O((S^c)^2)$ arithmetic operations. The N th term in the series will be of the order

$$O\left(\frac{T^{N+2}}{(N+2)!} N^2 \lambda_{\max}^N\right) \approx O\left(\frac{(T \lambda_{\max})^N}{(N)!}\right)$$

and it will take approximately $|T \lambda_{\max}|$ terms before the successive terms in the series start diminishing and definitely many more terms before the series can be truncated.

Apart from the high computational burden, the evaluation of series sum poses serious accuracy problems. The generator matrix is negative semi-definite, and its maximum eigenvalue may lie far off on the negative real axis. The growth of series terms is exponential, which, depending on $|T \lambda_{\max}|$, may lead to arithmetic overflow before the series converges. Even when overflow does not occur, the series has terms of

order $O\left(\frac{(T \lambda_{\max})^N}{N!}\right)$, which becomes several orders of magnitudes larger than T^2 , which is the maximum possible value of the joint statistics. The series, with increasingly large alternating terms, is supposed to sum to a small positive value, but due to finite word length of digital computers, the series sum will be grossly in error. To alleviate these problems, we propose a closed-form solution for the joint statistics in the following theorem.

Theorem 2: The series solution for the joint statistics, as proposed in **Theorem 1**, can be expressed as

$$E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\} = [\Phi C_j \Psi]_{kj'} + [\Phi C_{j'} \Psi]_{kj}$$

where elements of matrix C_j (and $C_{j'}$) can be expressed as explicit functions of eigensolution

$$[C_j]_{rc} = \psi_{rj} \phi_{jc} S_{rc} \begin{cases} \frac{T^2}{2} & \text{if } \lambda_r = \lambda_c = 0 \\ \frac{1 - 2e^{T\lambda}}{\lambda^2} & \text{if } \lambda_r = \lambda_c = \lambda \neq 0 \\ \frac{e^{T\lambda} - (1 + T\lambda)}{\lambda^2} & \text{if } \lambda_r = \lambda \neq 0; \lambda_c = 0 \\ \text{or, if } \lambda_c = \lambda \neq 0; \lambda_r = 0 \\ \frac{e^{T\lambda_r} - 1}{\lambda_r(\lambda_r - \lambda_c)} + \frac{e^{T\lambda_c} - 1}{\lambda_c(\lambda_c - \lambda_r)} & \text{if } \lambda_r \neq \lambda_c; \lambda_r, \lambda_c \neq 0 \end{cases}$$

Proof: See the Appendix.

Remark: We would expect that irrespective of initial state k , the finite-time joint statistics exhibit the following limiting behavior

$$\lim_{T \rightarrow \infty} \frac{E\{\tau_j(T)\tau_j(T) \mid y(0) = k\}}{T^2} = \pi_j \pi_j.$$

To show that the closed-form expression for the finite time joint statistics in **Theorem 2** indeed exhibits this behavior, we first note that

$$\lim_{T \rightarrow \infty} \frac{S_{rc}}{T^2} = \begin{cases} \frac{1}{2} & \text{if } \lambda_r = \lambda_c = 0 \\ 0 & \text{otherwise.} \end{cases}$$

In addition, exactly one eigenvalue of generator Q is zero. Let λ_i be that eigenvalue. Choose corresponding right-eigenvector $\phi_i = 1$. The left eigenvector must be $\Psi_i^T = \pi$. In terms of elements of matrices Φ and Ψ , we have $\phi_{ki} = 1$, $\psi_{ik} = \pi_k$; $\forall k$. Clearly, all but the i th diagonal element of matrices C_j and C_j vanish. Hence

$$\begin{aligned} \lim_{T \rightarrow \infty} \frac{E\{\tau_j(T)\tau_j(T) \mid y(0) = k\}}{T^2} &= \lim_{T \rightarrow \infty} \frac{\phi_{ki}[C_j]_{ii}\psi_{ij} + \phi_{ki}[C_j]_{ii}\psi_{ij}}{T^2} \\ &= \phi_{ki} \left(\psi_{ij}\phi_{ji} \frac{1}{2} \right) \psi_{ij} + \phi_{ki} \left(\psi_{ij}\phi_{ji} \frac{1}{2} \right) \psi_{ij} \\ &= \pi_j \frac{1}{2} \pi_j + \pi_j \frac{1}{2} \pi_j = \pi_j \pi_j. \end{aligned}$$

We now turn to finite time statistics $E\{\tau_j(T) \mid y(0) = k\}$. The following result, which is easy to prove, can be used to compute the conditional occupation times.

Theorem 3: Let $y(t)$ be a continuous-time Markov chain with generator $Q = \Phi \begin{bmatrix} 0 & \\ & \tilde{\Lambda} \end{bmatrix} \Psi$. Then, the occupation times in states j conditioned on initial state $y(0)$ is given by

$$\begin{aligned} E\{\tau_j(T) \mid y(0) = k\} &= \left[\sum_{N=0}^{\infty} \frac{T^{N+1}}{(N+1)!} \sum_{\forall l, m \mid l+m=N} (Q^l \delta_j Q^m) \mathbf{1} \right]_k \\ &= \left[\Phi \begin{bmatrix} T & \\ & \tilde{\Lambda} - 1(e^{T\tilde{\Lambda}} - 1) \end{bmatrix} \Psi \right]_{kj}. \end{aligned}$$

Proof: Identical to that for **Theorem 3** and outlined in [12].

IV. SUMMARY OF COMPUTATIONAL STEPS

We summarize below the computational steps involved in implementing the coordinating control. Discussion of simulation results in Section VI will further clarify many computation issues.

1) *Off-line Computations:* For each cell, construct the generator matrix using the failure-repair history of machines. Compute the finite-time and steady-state statistics using **Theorem 2** and **3**. Store the result for on-line computation.

2) *On-line Computations:* To specify the production rate

for the i th period of an N -period problem, the following steps are needed:

Step 1) Initialization: Set $P_N = -C_N$; $K_N = Q_N$

Step 2) For $k = (N - 1)$ to $(i - 1)$ repeat

 Compute F_k, M_k, P_k using (7) and steady-state values for simple and joint statistics.

Step 3) Observe the current cell states. Compute F_i, L_i, M_i using (7) and transient values for simple and joint statistics.

Step 4) Observe the current inventory x_i and compute production rates u_i , using (6).

Observe that *Step 1)* and *2)* can be performed during the coordination period. At the coordination epoch, cell and inventory states are observed, and production rates are specified using *Steps 3)* and *4)*, which involves very little computation. The coordinator, in effect, is able to respond instantaneously after receiving the feedback, which is a requirement for real-time control.

V. THE MYOPIC LINEAR PROGRAMMING FORMULATION

For the purposes of comparison, we consider here a myopic linear programming formulation of the problem as an alternative to the stochastic dynamic programming formulation developed in Section II. Linear programming has been used frequently in the operations research/management science literature for capacity allocation across part types. The time-scale of decision making, in such models, is of the order of a shift to a week, and the objective is to plan aggregate production based on the expected availability of resources. In contrast, we are concerned with the real-time control decision, which has to be taken on a time-scale comparable to the processing times.

The linear programming formulation for capacity allocation in any period is based on the expected availability of machines conditioned on the state at the beginning of that period. Its performance will provide a yardstick against which we can measure the performance of the stochastic dynamic programming formulation, which uses not only expected availability, but also the higher order statistics of machine states. The LP formulation is myopic in the sense that production rates for each period are specified based on beginning inventory and demand during that period alone. The problem is solved independently for each period, but the beginning inventory has all the information needed from the earlier periods. In effect, the cell controller behaves like an open-loop feedback controller, which solves the one period problem N times, using the information x_k available thus far each time, but behaving as if no further demand will take place in the future.

Let

- $h_k(i)$ inventory holding cost for product i for period k
- $b_k(i)$ inventory backlogging cost for product i for period k
- $\bar{x}_{k+1}(i)$ expected inventory for product i at the end of k th period or, equivalently, at the beginning of $(k + 1)$ th period.

$$\bar{x}_{k+1}^+(i) = \max(0, \bar{x}_{k+1}(i));$$

$$\bar{x}_{k+1}^-(i) = \max(0, -\bar{x}_{k+1}(i));$$

$$h_k = [h_k(1) \cdots h_k(p)]; b_k = [b_k(1) \cdots b_k(p)];$$

$$\bar{x}_{k+1}^+ = \begin{bmatrix} \bar{x}_{k+1}^+(1) \\ \vdots \\ \bar{x}_{k+1}^+(p) \end{bmatrix}; \bar{x}_{k+1}^- = \begin{bmatrix} \bar{x}_{k+1}^-(1) \\ \vdots \\ \bar{x}_{k+1}^-(p) \end{bmatrix};$$

$$\bar{x}_{k+1} = \begin{bmatrix} \bar{x}_{k+1}(1) \\ \vdots \\ \bar{x}_{k+1}(p) \end{bmatrix}.$$

To specify production rates for k th period, the cell coordinator solves the following linear program

$$\begin{aligned} &\text{Minimize } h_k \bar{x}_{k+1}^+ + b_k \bar{x}_{k+1}^- \\ &\text{Subject to } Du_k \leq a \\ &\quad \bar{x}_{k+1} = x_k + \bar{B}_k u_k - d_k \\ &\quad \bar{x}_{k+1}^+ = \bar{x}_{k+1}^+ - \bar{x}_{k+1}^- \\ &\quad \bar{x}_{k+1}^+, \bar{x}_{k+1}^- \geq 0; u_k \geq 0. \end{aligned}$$

The above linear program computes production rates in such a way that expected holding and backloging cost is minimized. By incorporating costs for a number of periods, the formulation can be easily extended to allow a limited "look ahead."

VI. SIMULATION RESULTS

In this section, we compare the effectiveness of the proposed control rules using a simple example. Our purpose is to give the reader some additional insight into how the proposed control laws achieve the production target in the face of machine failures and repairs.

Consider a manufacturing system that consists of two parallel cells where each has two types of machines. Cell 1 has five each of Type I and Type II machines, whereas Cell 2 consists of six of Type III and four of Type IV machines. The MTBF and MTTR of these machines are listed in Table I. We consider a five-period problem with two products with time varying demands as indicated in Table II. The manufacturing system runs 8 hr every period. The two products can be processed in either cell, and their processing times are listed in Table III. Note that the load on the system, based on the expected availability of machines, varies from underutilized capacities in period 1 to a shortage of capacities in period 3. Moreover, the five-period average load is roughly matched to the expected capacity. To illustrate this, first note that, on an average, machine type I is available for $(5)(8)\left(\frac{10}{10+1.6}\right) = 34.5$ hr. Similarly machine types II, III, and IV are available for 35.6, 41.4, and 28.4 hours of processing, respectively. The average demand, on the other hand, is 380 units for product 1 and 640 units for product 2. Consider a deterministic allocation problem where every period, machines are functional for exactly the durations of their average availabilities. Is it possible to meet the average demands? Suppose a fraction f_i of product i ($0 \leq f_i \leq 1$;

TABLE I
MACHINE PARAMETERS

Cell	Machine Type	MTBF	MTTR
1	Type I	10 Hours	1.6 Hour
	Type II	8 Hours	1 Hour
2	Type III	10 Hours	1.6 Hour
	Type IV	8 Hours	1 Hour

TABLE II
DEMAND

Period	1	2	3	4	5
Product 1	300	400	500	400	300
Product 2	500	700	800	700	500

TABLE III
PROCESSING TIMES

Product	Cell 1		Cell 2	
	Machine Type I	Machine Type II	Machine Type III	Machine Type IV
1	0.05 Hour	0.066 Hour	0.05 Hour	0.1 Hour
2	0.066 Hour	0.05 Hour	0.1 Hour	0.05 Hour

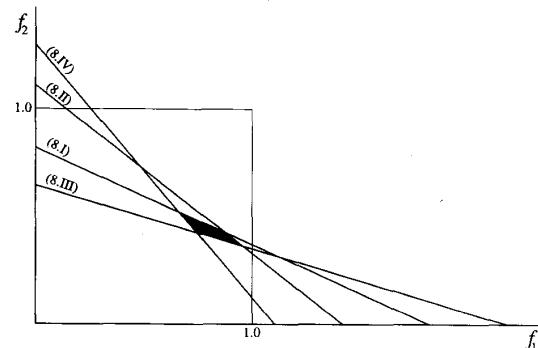


Fig. 4. Feasible allocation based on average capacities and demand.

$i = 1, 2$) is assigned to cell 1. Then, the average demands would be met if allocations f_i 's satisfy capacity constraints

$$(380)(0.05)f_1 + (640)(0.066)f_2 \leq 34.5 \quad (8a)$$

$$(380)(0.066)f_1 + (640)(0.05)f_2 \leq 35.6 \quad (8b)$$

$$(380)(0.05)(1-f_1) + (640)(0.1)(1-f_2) \leq 41.4 \quad (8c)$$

$$(380)(0.1)(1-f_1) + (640)(0.05)(1-f_2) \leq 28.4. \quad (8d)$$

These constraints are shown in Fig. 4, where shaded region represents feasible allocations. It appears in this case that, on average, demand can be met. Recall, however, that constraints set changes stochastically, and a feasible allocation based on average capacity may be infeasible for many realizable machine states. In fact, the allocation has to be changed as capacity changes. The feasibility based on averages does not have much meaning except that it conveys, to a limited extent, the potential load on the system. Infeasibility of constraint set in Fig. 4, for example, would mean a heavily loaded system. Small feasible set can be interpreted as roughly matched capacity. The actual output will depend on how the allocation is achieved dynamically.

A cell typically consists of a large number of machines of different types. The machine state grows exponentially in the number of machines. However, many machine states are extremely unlikely to be reached, and there is no need to compute production rates corresponding to those improbable

TABLE IV
STATIONARY PROBABILITIES FOR DOMINANT STATES OF CELL 1

Machine State	(6,4)	(5,4)	(4,4)	(3,4)	(6,3)	(5,3)	(4,3)	(6,2)	(5,2)
Stationary Probability	0.256	0.246	0.098	0.021	0.128	0.123	0.049	0.024	0.023

TABLE V
STATIONARY PROBABILITIES FOR DOMINANT STATES OF CELL 2

Machine State	(5,5)	(4,5)	(3,5)	(5,4)	(4,4)	(3,4)	(5,3)	(4,3)
Stationary Probability	0.264	0.211	0.068	0.165	0.132	0.042	0.041	0.033

states. Cell 1, in this example, can have a total of $(6 + 1)(4 + 1) = 35$ possible machine states. However, an examination of stationary probabilities of the associated Markov chain shows that Cell 1 is in one of the nine dominant states listed in Table IV 97% of the time. Similarly, out of a total of 36 possible states, Cell 2 is in one of the eight states listed in Table V 96% of the time. The unlikely states correspond to the situation when a large number of machines are under repair; this situation is extremely unlikely to happen in any realistic manufacturing system with a reasonable maintenance policy.

For each cell, the matrix $\frac{1}{T} \tau(T)$ is computed as a function of trial coordination period T using **Theorem 3**. The computations indicate that for each cell, as T approaches 8 hr, the rows of matrix $\frac{1}{T} \tau(T)$ converge to a unique vector, which is the stationary probability vector. This allows us to consider a coordination period equal to 8 hr so that the cell coordinator computes production rates once every period. At the end of each period, the demand is met from the available inventory, and costs are incurred on any extra inventory or backlogged demand. We assume that product 1 is more expensive than product 2 to hold and backlog. More specifically, a cost of 5 units is incurred for each unit of product 1, whereas a cost of only 1 unit is incurred for each unit of product 2 whenever ending inventory exceeds or falls short of demand. The elements of Q_k , R_k , and C_k are design parameters that should be chosen to fit the cost structure of a specific problem. Since positive and negative inventory cost the same and product 1 inventory is five times as expensive as that of product 2, we have taken $Q_k = \text{diag}[5 \ 1]$, and $C_k = O$. In addition, for this example, we used $R_k = \text{diag}[5 \ \dots \ 5]$.

We simulated the manufacturing system on a VAXstation using IMSL code for matrix manipulation and random number generation. The two decision rules were imbedded inside the discrete event simulator so that production rates could be computed based on the sample path realization thus far. We found occasional violation of capacity constraint (4) by the stochastic dynamic program, which we corrected heuristically by scaling the production rates of both the products. In addition, any negative production rates computed by the stochastic dynamic program were set to zero. This rarely occurs in a heavily loaded system, which we are considering here. The same sample path realizations of machine failures/repairs were used for stochastic dynamic program

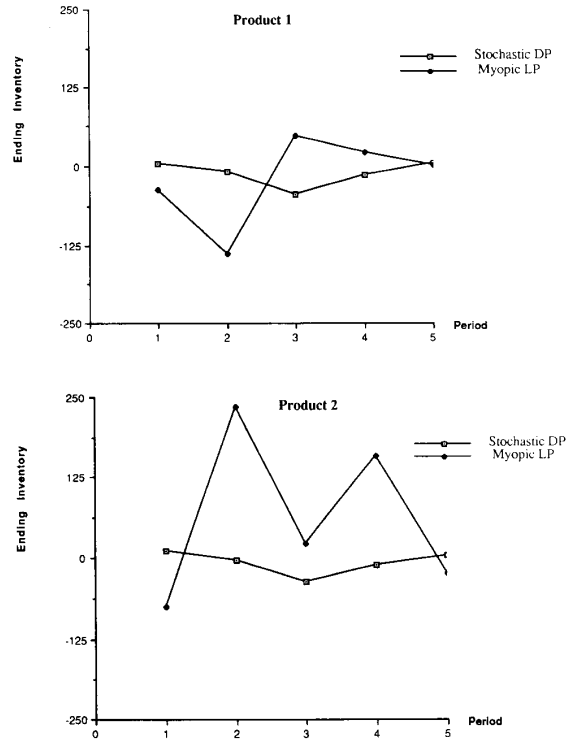


Fig. 5. Sample path of ending inventories.

as well as a myopic linear programming solution so that the sample paths of inventory levels can be compared under the two decision rules. The simulation was always started with zero inventory of both products.

It was found that the sample path of inventory levels under the stochastic dynamic program was smoother than that under myopic linear program. Two representative sample realizations are shown in Figs. 5 and 6. These realizations show that the deviation from the mean is smaller under the stochastic dynamic program than with the myopic linear program. A more detailed analysis, based on 25 random experiments, is described in Tables VI and VII. The average ending inventory levels under the two decision rules are not very different. However, their standard deviation and range of variation is significantly different. Comparisons based on average inventories can be very misleading. A low average inventory over the samples may seem to perform well, but if it is accompanied with high variance, it will incur high holding and backlogging costs. It is due to this reason that stochastic dynamic program yields much lower costs compared with myopic linear program as shown in Table VIII. In addition, in today's manufacturing environment, where reliability of supply is extremely important, it is not sufficient that we meet the demand on the average. An equally important question asks: How far do we end up from the targeted production? It is due to this reason that small standard deviations and ranges of variation of ending inventories of stochastic dynamic program makes it a very attractive control policy. In fact, the impact of supply variance increases dramatically in a multi-stage or multiplant environment. Thus, the variance reduc-

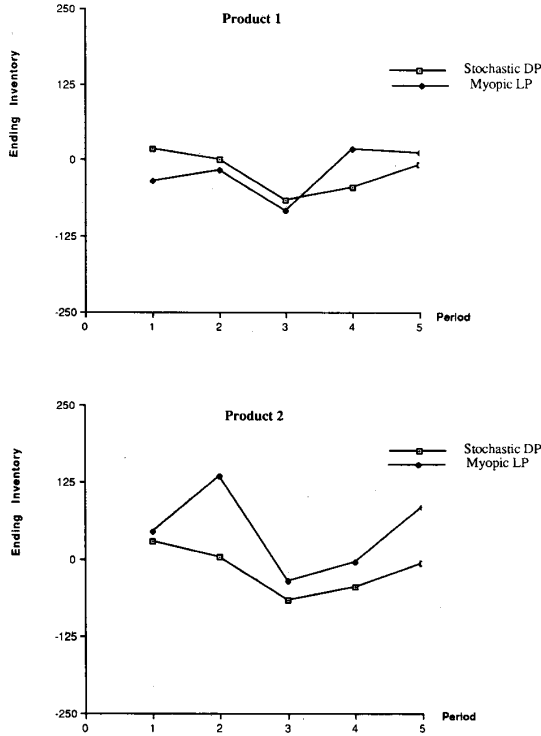


Fig. 6. Another sample path of ending inventories.

TABLE VI
ENDING INVENTORY OF PRODUCT 1

Period	Stochastic Dynamic Program				Myopic Linear Program			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min	Max.
1	8	7	-2	25	3	77	-149	151
2	-12	14	-65	2	-7	99	-202	156
3	-63	28	-126	-22	6	80	-121	167
4	-40	25	-98	-3	-6	85	-188	144
5	-3	12	-39	12	32	82	-129	237

TABLE VII
ENDING INVENTORY OF PRODUCT 2

Period	Stochastic Dynamic Program				Myopic Linear Program			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min	Max.
1	18	7	8	35	21	107	-222	222
2	-6	13	-51	7	28	135	-149	343
3	-58	26	-125	-16	-27	94	-209	172
4	-37	25	-86	-2	42	93	-120	275
5	-2	12	-37	13	2	86	-150	144

tion achieved at the output by the proposed policy has a major beneficial impact on reducing the total systemic cost in a multistage system.

VII. CONCLUSIONS

We have developed a real-time dispatch algorithm for multicell coordination. The algorithm is responsive to machine failures in systems where the host and local cell controllers communicate periodically. The algorithm has been tested via simulation. It has the appealing characteristic that the resulting production is such that variation from demand is kept small. In contrast with previous approaches, the large

TABLE VIII
HOLDING AND BACKLOGGING COSTS

Costs	Stochastic Dynamic Program				Myopic Linear Program			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min	Max.
Holding	78	45	8	171	1158	608	71	2070
Backlogging	716	386	232	1589	947	480	136	2240
Total	794	366	309	1598	2105	721	912	4241

machine state space can still be handled. There are several other possible variations of this algorithm. For instance, another possible approach is demand allocation to cells by the host, followed by cell controllers that try to achieve production targets by responding to machine states. A stricter imposition of capacity constraints in developing the analysis is yet another possibility.

We observe that the key concept we used here was that of aggregation, that is, the higher level estimates, in an aggregate way, the expected availability of capacity in different machine states. This includes the first- and second-order statistics information. The lower level controller implements the policy based on this information in real time. In a single cell context, the two levels are implemented by using a single control computer on two different time scales.

Finally, note that the result on the finite-time second-order statistics is general and could be useful in a wide variety of situations.

APPENDIX

Proof of Theorem 1: To evaluate the finite time joint statistics of the random variables $\tau_j(T)$ and $\tau_{j'}(T)$, which are the occupation times in state j and j' , respectively, we define the Laplace transform associated with their joint distribution conditioned on the initial state $y(0) = k$. This is given by two-dimensional

$$f_{jj'}(\sigma_1, \sigma_2, T, k) = E\{e^{-\sigma_1\tau_j(T) - \sigma_2\tau_{j'}(T)} | y(0) = k\}$$

$$= E^k\{e^{-\sigma_1\tau_j(T) - \sigma_2\tau_{j'}(T)}\}.$$

All moments of the joint distribution can then be computed from the transform function. In particular, the first moment is

$$E\{\tau_j(T)\tau_{j'}(T) | y(0) = k\}$$

$$= \frac{\partial}{\partial \sigma_1} \frac{\partial}{\partial \sigma_2} f_{jj'}(\sigma_1, \sigma_2, T, k) |_{\sigma_1 = \sigma_2 = 0}. \quad (9)$$

To obtain the transform function, we develop a linear differential equation for it, where for notational simplicity, we suppress explicit dependence of $f_{jj'}(\sigma_1, \sigma_2, T, k)$ on the transform variables σ_1 , and σ_2 by defining

$$f_{jj'}(T, k) = f_{jj'}(\sigma_1, \sigma_2, T, k).$$

In addition, define vector

$$f_{jj'}(T) = \begin{bmatrix} f_{jj'}(T, 1) \\ \vdots \\ f_{jj'}(T, S^c) \end{bmatrix}$$

where S^c is the number of machine states. With this notation, $[f_{jj'}(T)]_k = f_{jj'}(T, k)$ and the initial state $f_{jj'}(0) = 1$.

In order to develop the differential equation for $f_{jj'}(T, k)$ consider

$$\begin{aligned} f_{jj'}(T + \Delta, k) &= E^k \{ e^{-\sigma_1 \tau_j(T+\Delta) - \sigma_2 \tau_{j'}(T+\Delta)} \} \\ &= E^k \{ e^{-\sigma_1(\tau_j(T+\Delta) - \tau_j(\Delta)) - \sigma_2(\tau_{j'}(T+\Delta) - \tau_{j'}(\Delta))} \\ &\quad \cdot e^{-\sigma_1 \tau_j(\Delta) - \sigma_2 \tau_{j'}(\Delta)} \} \end{aligned}$$

Conditioning on $y_\Delta = y(\Delta)$, which is the state at time Δ , we get

$$\begin{aligned} f_{jj'}(T + \Delta, k) &= E^k \{ e^{-\sigma_1 \tau_j(\Delta) - \sigma_2 \tau_{j'}(\Delta)} \\ &\quad E^{y_\Delta} \{ e^{\sigma_1(\tau_j(T+\Delta) - \tau_j(\Delta)) - \sigma_2(\tau_{j'}(T+\Delta) - \tau_{j'}(\Delta))} | F_\Delta \} \} \end{aligned}$$

where F_Δ is σ algebra generated by $\{y(t), 0 \leq t \leq \Delta\}$. Expanding the first exponential, discarding terms of higher order than $O(\Delta)$, and using Markovian property of memoryless yields

$$\begin{aligned} f_{jj'}(T + \Delta, k) &\approx E^k \{ (1 - \sigma_1 \tau_j(\Delta) \\ &\quad - \sigma_2 \tau_{j'}(\Delta)) f_{jj'}(T, y_\Delta) \} \\ &= \sum_{i=1}^{S^c} P_{ki}(\Delta) f_{jj'}(T, i) \\ &\quad \cdot (1 - \sigma_1 \tau_j(\Delta) - \sigma_2 \tau_{j'}(\Delta)) \quad (10) \end{aligned}$$

where $P_{ki}(\Delta) = Pr\{y(\Delta) = i | y(0) = k\}$. Note that $\tau_j(\Delta)$ and $\tau_{j'}(\Delta)$ are the times spent in states j and j' during the infinitesimal interval Δ , which depend on the initial state k and are given by

$$\tau_j(\Delta) = \Delta \delta_{jk} = \begin{cases} \Delta & \text{if } j = k, \\ 0 & \text{otherwise.} \end{cases}$$

Defining

$$\begin{aligned} \lim_{\Delta \rightarrow 0^+} \frac{1 - P_{kk}(\Delta)}{\Delta} &= q_{kk} \\ \lim_{\Delta \rightarrow 0^+} \frac{P_{ki}(\Delta)}{\Delta} &= q_{ki}, \quad k \neq i \end{aligned}$$

which are the transition rates of the continuous-time Markov chain, which can be computed from the MTBF and MTTR of the machines. Disregarding the second-order terms, (10) can now be written as

$$\begin{aligned} f_{jj'}(T + \Delta, k) &= \sum_{i=1, i \neq k}^{S^c} q_{ki} \Delta f_{jj'}(T, i) (1 - \sigma_1 \Delta \delta_{jk} - \sigma_2 \Delta \delta_{j'k}) \\ &\quad + (1 - q_{kk} \Delta) f_{jj'}(T, k) (1 - \sigma_1 \Delta \delta_{jk} - \sigma_2 \Delta \delta_{j'k}) \\ &= \sum_{i=1, i \neq k}^{S^c} q_{ki} \Delta f_{jj'}(T, i) - q_{kk} \Delta f_{jj'}(T, k) \\ &\quad - \sigma_1 \Delta \delta_{jk} f_{jj'}(T, k) - \sigma_2 \Delta \delta_{j'k} f_{jj'}(T, k) \\ &\quad + f_{jj'}(T, k). \end{aligned}$$

Stacking similar expressions for $k = 1, 2, \dots, S^c$ leads to

$$\begin{aligned} f_{jj'}(T + \Delta) &= Q \Delta f_{jj'}(T) \\ &\quad - \sigma_1 \Delta \delta_j f_{jj'}(T) - \sigma_2 \Delta \delta_{j'} f_{jj'}(T) + f_{jj'}(T) \\ \text{or } \frac{f_{jj'}(T + \Delta) - f_{jj'}(T)}{\Delta} &= (Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) f_{jj'}(T) \quad (11) \end{aligned}$$

where,

$$\begin{aligned} Q &= \begin{bmatrix} -q_{11} & q_{12} & \dots & q_{1S^c} \\ q_{21} & -q_{22} & \dots & q_{2S^c} \\ \vdots & \vdots & \ddots & \vdots \\ q_{S^c 1} & q_{S^c 2} & \dots & -q_{S^c S^c} \end{bmatrix}, \text{ and} \\ \delta_j &= \begin{bmatrix} \delta_{j1} & & & \\ & \delta_{j2} & & \\ & & \ddots & \\ & & & \delta_{jS^c} \end{bmatrix}. \end{aligned}$$

Taking limit $\Delta \rightarrow 0$, (11) yields the matrix differential equation

$$\begin{aligned} \lim_{\Delta \rightarrow 0} \frac{f_{jj'}(T + \Delta) - f_{jj'}(T)}{\Delta} &= \frac{\partial f_{jj'}(T)}{\partial T} \\ &= (Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) f_{jj'}(T) \end{aligned}$$

whose solution is

$$f_{jj'}(T) = e^{(Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) T} f_{jj'}(0)$$

which, when substituted in (9), gives

$$\begin{aligned} E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\} &= \left[\frac{\partial}{\partial \sigma_1} \frac{\partial}{\partial \sigma_2} e^{(Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) T} f_{jj'}(0) \Big|_{\sigma_1 = \sigma_2 = 0} \right]_k. \end{aligned}$$

To evaluate the above expression, one could use the series definition for the matrix exponential

$$e^{(Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) T} = I + \sum_{k=1}^{\infty} \frac{(Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'})^k T^k}{k!}$$

and collect the coefficients of $\sigma_1 \sigma_2$. Other terms vanish when expression

$$\frac{\partial}{\partial \sigma_1} \frac{\partial}{\partial \sigma_2} e^{(Q - \sigma_1 \delta_j - \sigma_2 \delta_{j'}) T} f_{jj'}(0) \Big|_{\sigma_1 = \sigma_2 = 0}$$

is evaluated. Observe that all coefficients of $\sigma_1 \sigma_2$ have only one δ_j and $\delta_{j'}$ term each, in different orders, and with powers of Q between and on either side. Corresponding to the i th term in the expansion, the powers of Q enumerate all possible partitions of $(i - 2)$. The final result is

$$\begin{aligned} E\{\tau_j(T) \tau_{j'}(T) | y(0) = k\} &= \left[\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \sum_{\forall i, l, m | i+l+m=N} \right. \\ &\quad \left. \cdot (Q^i \delta_j Q^l \delta_{j'} Q^m + Q^i \delta_{j'} Q^l \delta_j Q^m) f_{jj'}(0) \right]_k \end{aligned}$$

where $f_{jj'}(0) = \mathbf{1}$ from initial condition. ■

Proof of Theorem 2: First, we observe that $\mathbf{1}$ is an eigenvector of generator matrix \mathbf{Q} corresponding to eigenvalue 0. This implies $\mathbf{Q}^m \mathbf{1} = \mathbf{0}$ for all $m \neq 0$. Thus, in the series solution, terms corresponding to $m \neq 0$ can be eliminated yielding

$$\begin{aligned} E\{\tau_j(T)\tau_{j'}(T) | y(0) = k\} \\ &= \left[\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \right. \\ &\quad \left. \sum_{\forall i, l | i+l=N} (\mathbf{Q}^i \delta_j \mathbf{Q}^l \delta_{j'} + \mathbf{Q}^i \delta_{j'} \mathbf{Q}^l \delta_j) \mathbf{1} \right]_k \\ &= \left[\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \sum_{\forall i, l | i+l=N} \mathbf{Q}^i \delta_j \mathbf{Q}^l \right]_{kj'} \\ &\quad + \left[\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \sum_{\forall i, l | i+l=N} \mathbf{Q}^i \delta_{j'} \mathbf{Q}^l \right]_{kj} \end{aligned} \quad (12)$$

The first term in (12) is an element of the matrix

$$\begin{aligned} &\sum_{N=0}^{\infty} \frac{T^{N+2}}{(N+2)!} \sum_{\forall i, l | i+l=N} \mathbf{Q}^i \delta_j \mathbf{Q}^l \\ &= \sum_{i=0}^{\infty} \mathbf{Q}^i \delta_j \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \mathbf{Q}^l \\ &= \sum_{i=0}^{\infty} \Phi \Lambda^i \Psi \delta_j \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \Phi \Lambda^l \Psi \\ &= \sum_{i=0}^{\infty} \Phi \Lambda^i \Psi \delta_j \Phi D^{(i)} \Psi \end{aligned} \quad (13)$$

where diagonal matrix $D^{(i)}$ is defined as

$$D^{(i)} = \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \Lambda^l = \begin{bmatrix} d_1^{(i)} & & & \\ & d_2^{(i)} & & \\ & & \ddots & \\ & & & d_{sc}^{(i)} \end{bmatrix};$$

$$d_c^{(i)} = \begin{cases} \frac{T^{i+2}}{(i+2)!} & \text{if } \lambda_c = 0 \\ \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \lambda_c^l \\ \frac{e^{T\lambda_c} - \sum_{\nu=0}^{\nu=i+1} \frac{(T\lambda_c)^\nu}{\nu!}}{\lambda_c^{i+2}} & \text{if } \lambda_c \neq 0. \end{cases}$$

Matrix (13) can be further simplified to

$$\sum_{i=0}^{\infty} \Phi \Lambda^i \Psi \delta_j \Phi D^{(i)} \Psi = \Phi \left(\sum_{i=0}^{\infty} \Lambda^i \Psi \delta_j \Phi D^{(i)} \right) \Psi = \Phi C_j \Psi. \quad (14)$$

Substituting (14) in (12) we have

$$E\{\tau_j(T)\tau_{j'}(T) | y(0) = k\} = [\Phi C_j \Psi]_{kj'} + [\Phi C_{j'} \Psi]_{kj}.$$

Thanks to the diagonal structure of matrices Λ^i and $D^{(i)}$ and the special structure of δ_j , the elements of matrix C_j can be expressed as

$$[C_j]_{rc} = \left[\sum_{i=0}^{\infty} \Lambda^i \Psi \delta_j \Phi D^{(i)} \right]_{rc} = \psi_{rj} \phi_{jc} \sum_{i=0}^{\infty} \lambda_r^i d_c^{(i)} \quad (15)$$

which is in a much more tractable form than the multiple matrix product in the parenthesis. However, the deceptively simple expression (15) still has the undesirable series form, and the task remains now to show that the infinite series can be expressed in the closed form. We take this task next. More specifically, we are going to show that

$$S_{rc} = \sum_{i=0}^{\infty} \lambda_r^i d_c^{(i)} = \begin{cases} \frac{T^2}{2} & \text{if } \lambda_r = \lambda_c = 0 \\ \frac{1 - 2e^{T\lambda}}{\lambda^2} & \text{if } \lambda_r = \lambda_c = \lambda \neq 0 \\ \frac{e^{T\lambda} - (1 + T\lambda)}{\lambda^2} & \text{if } \lambda_r = \lambda \neq 0; \lambda_c = 0 \\ \text{or, if } \lambda_c = \lambda \neq 0; \lambda_r = 0 \\ \frac{e^{T\lambda_r} - 1}{\lambda_r(\lambda_r - \lambda_c)} + \frac{e^{T\lambda_c} - 1}{\lambda_c(\lambda_c - \lambda_r)} & \text{if } \lambda_r \neq \lambda_c; \lambda_r, \lambda_c \neq 0 \end{cases}$$

Consider each case separately:

1) $\lambda_r = \lambda_c = 0$.

The series has only one nonzero term and the result follows trivially.

2) $\lambda_r = \lambda_c = \lambda \neq 0$.

Substituting for $d_c^{(i)}$ gives

$$\begin{aligned} S_{rc} &= \sum_{i=0}^{\infty} \lambda^i \frac{e^{T\lambda} - \sum_{\nu=0}^{\nu=i+1} \frac{(T\lambda)^\nu}{\nu!}}{\lambda^{i+2}} \\ &= \frac{1}{\lambda^2} \sum_{i=0}^{\infty} \left(e^{T\lambda} - \sum_{\nu=0}^{\nu=i+1} \frac{(T\lambda)^\nu}{\nu!} \right) \end{aligned}$$

The above infinite series has a structure that will become evident if the terms are grouped in the following way. Collect the last term corresponding to each i to form an infinite series. Next, collect the second-to-last term corresponding to each i to form another infinite series. Continue this process by taking third-to-last terms, fourth-to-last terms, etc., for each i (substitute zero if no such term exists) and forming a

new infinite series every time. The result is

$$\begin{aligned}
 S_{rc} &= \frac{1}{\lambda^2} \left(- \sum_{\nu=1}^{\infty} \frac{(T\lambda)^\nu}{\nu!} \right) + \frac{1}{\lambda^2} \left(- \sum_{\nu=0}^{\infty} \frac{(T\lambda)^\nu}{\nu!} \right) \\
 &+ \frac{1}{\lambda^2} \left(e^{T\lambda} - \sum_{\nu=0}^{\infty} \frac{(T\lambda)^\nu}{\nu!} \right) \\
 &+ \frac{1}{\lambda^2} \left(e^{T\lambda} - \sum_{\nu=0}^{\infty} \frac{(T\lambda)^\nu}{\nu!} \right) + \dots \\
 &= \frac{1}{\lambda^2} (1 - e^{T\lambda}) + \frac{1}{\lambda^2} (-e^{T\lambda}) \\
 &+ 0 + 0 + \dots = \frac{1 - 2e^{T\lambda}}{\lambda^2}.
 \end{aligned}$$

- 3) $\lambda_r = \lambda \neq 0$; $\lambda_c = 0$ or, $\lambda_c = \lambda \neq 0$; $\lambda_r = 0$. The result can be obtained by direct substitution.
- 4) $\lambda_r \neq \lambda_c$; $\lambda_r, \lambda_c \neq 0$.

We first observe that

$$\begin{aligned}
 S_{rc} &= \sum_{i=0}^{\infty} \lambda_r^i d_c^{(i)} \\
 &= \sum_{i=0}^{\infty} \lambda_r^i \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \lambda_c^l \\
 &= \sum_{l=0}^{\infty} \lambda_c^l \sum_{i=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \lambda_r^i \\
 &= S_{cr}.
 \end{aligned}$$

This shows that series sum S_{cr} is symmetric with respect to r and c . Now, we turn to the task of evaluating the sum

$$\begin{aligned}
 S_{cr} &= \sum_{i=0}^{\infty} \lambda_r^i \sum_{l=0}^{\infty} \frac{T^{i+l+2}}{(i+l+2)!} \lambda_c^l \\
 &= \sum_{i=0}^{\infty} \lambda_r^i \frac{e^{T\lambda} - \sum_{\nu=0}^{\nu=i+1} \frac{(T\lambda_c)^\nu}{\nu!}}{\lambda_c^{i+2}}.
 \end{aligned}$$

The above infinite series again has a simplifying structure that will become evident if the terms are regrouped in the following manner. Collect the first term corresponding to each i to form an infinite series. Next, collect the second term corresponding to each i to form another infinite series. Continue this process by taking third terms, fourth terms, etc., for each i (substitute zero if no such term exists) and forming a new infinite series every time. We get

$$\begin{aligned}
 S_{rc} &= \frac{e^{T\lambda_c}}{\lambda_c^2} \sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i - \frac{1}{\lambda_c^2} \sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i \\
 &- \frac{1}{\lambda_c} T \sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i \\
 &- \frac{1}{\lambda_c} \frac{T^2 \lambda_r}{2!} \sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i
 \end{aligned}$$

$$\begin{aligned}
 &- \frac{1}{\lambda_c} \frac{T^3 \lambda_r^2}{3!} \sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i - \dots \\
 &= \frac{e^{T\lambda_c} - 1}{\lambda_c^2} \left(\sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i \right) \\
 &- \frac{1}{\lambda_c} \left(\sum_{\nu=1}^{\infty} \frac{T^\nu \lambda_r^{\nu-1}}{\nu!} \right) \left(\sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i \right).
 \end{aligned}$$

If $\lambda_c > \lambda_r$, the geometric series converges, and we have $\sum_{i=0}^{\infty} \left(\frac{\lambda_r}{\lambda_c} \right)^i = \frac{\lambda_c}{\lambda_c - \lambda_r}$, which gives

$$\begin{aligned}
 S_{rc} &= \frac{e^{T\lambda_c} - 1}{\lambda_c^2} \left(\frac{\lambda_c}{\lambda_c - \lambda_r} \right) - \frac{1}{\lambda_c} \left(\frac{e^{T\lambda_r} - 1}{\lambda_r} \right) \left(\frac{\lambda_c}{\lambda_c - \lambda_r} \right) \\
 &= \frac{e^{T\lambda_c} - 1}{\lambda_c(\lambda_c - \lambda_r)} + \frac{e^{T\lambda_r} - 1}{\lambda_r(\lambda_r - \lambda_c)}.
 \end{aligned}$$

If $\lambda_c < \lambda_r$, we use the symmetry property to obtain

$$\begin{aligned}
 S_{rc} &= S_{cr} \\
 &= \frac{e^{T\lambda_r} - 1}{\lambda_r^2} \left(\sum_{i=0}^{\infty} \left(\frac{\lambda_c}{\lambda_r} \right)^i \right) \\
 &- \frac{1}{\lambda_r} \left(\sum_{\nu=1}^{\infty} \frac{T^\nu \lambda_c^{\nu-1}}{\nu!} \right) \left(\sum_{i=0}^{\infty} \left(\frac{\lambda_c}{\lambda_r} \right)^i \right)
 \end{aligned}$$

which yields the same expression ■

ACKNOWLEDGMENT

We are grateful to the referees for many helpful comments. The authors are indebted to S. Shreve for suggesting the transform approach and contributing to **Theorem 1**. We would like to thank C. Abraham, B. Crowder, and R. Jayaraman of the Manufacturing Research at IBM's T. J. Watson Research Center group for having been extremely supportive over an extended period of time.

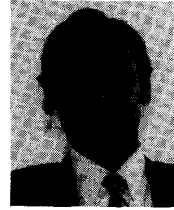
REFERENCES

- [1] R. Akella, M. R. Singh, and Y. Bassok, "Real-time part dispatch in flexible assembly, test and manufacturing systems," in *Artificial Intelligence: Manufacturing Theory and Practice* (S. Kumara, R. L. Kashyap, and A. Soyster, Eds). New York: Industrial Engineering and Management Institute of Industrial Engineers, 1989, pp. 297-338.
- [2] R. Akella and P. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 116-126, 1986.
- [3] M. R. Singh, "Planning and control decisions in manufacturing systems with uncertainties," Ph.D. dissertation, Grad. School Ind. Admn., Carnegie Mellon Univ., Pittsburgh, PA, 1990.
- [4] R. Akella, Y. Choong, and S. B. Gershwin, "Performance of a hierarchical production scheduling policy," *IEEE Trans. Components Hybrids Manuf. Technol.*, vol. GHMT-7, pp. 225-240, 1984.
- [5] D. Bertsekas *Dynamic Programming and Stochastic Control*. New York: Academic, 1976.
- [6] W. P. Blair and D. D. Sworder, "Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria," *Int. J. Contr.*, vol. 21, pp. 833-841, 1975.
- [7] H. J. Chizeck, A. S. Willsky, and D. Castanon, "Discrete-time Markovian-jump linear quadratic optimal control," *Int. J. Contr.*, vol. 43, pp. 213-231, 1986.

- [8] S. B. Gershwin, R. Akella, and Y. Choong, "Short-term production scheduling of an automated manufacturing facility," *IBM J. Res. Development*, vol. 29, pp. 392-400, 1985.
- [9] J. G. Kimemia and S. B. Gershwin, "An algorithm for the computer control of production in flexible manufacturing systems," *IIE Trans.*, vol. 15, pp. 353-362, 1983.
- [10] J. G. Shanthikumar and D. D. Yao, "Optimal allocation of buffers in a system of manufacturing cells," to be published.
- [11] J. G. Shanthikumar and D. D. Yao, "Optimal server allocation in a system of multi-server stations," to be published.
- [12] M. R. Singh, R. Akella, and B. Krogh, "Modeling failure-prone manufacturing systems as continuous-time Markov chain—Some computational issues," Working Paper 24-87-88, Grad. School Ind. Admin., Carnegie Mellon Univ., Pittsburgh, PA, 1987.
- [13] U. Sumita, J. G. Shanthikumar, and Y. Masuda, "Analysis of fault tolerant computer systems," *Microelectron. Rel.*, vol. 27, pp. 65-78, 1987.
- [14] D. D. Sworder, "Feedback control of a class of linear systems with jump parameters," *IEEE Trans. Automat. Contr.*, vol. AC-14, pp. 9-14, 1969.
- [15] W. M. Wohnam, "Random differential equations in control theory," in *Probabilistic Methods in Applied Mathematics, Vol. 2* (A. T. Bharucha-Reid, Ed.). New York: Academic, 1970, pp. 131-212.
- [16] D. D. Yao and J. G. Shanthikumar, "The optimal input rates to a system of manufacturing cells," to be published.
- [17] D. D. Yao and J. G. Shanthikumar, "Some resource allocation problems in multi-cell systems," in *Proc. 2nd TIMS/ORSA Conf. Flexible Manuf. Syst.* (Ann Arbor, MI), 1986, pp. 245-255.

For photograph and biography of **Ramakrishna Akella**, please see p. 607 of this TRANSACTIONS.

For photograph and biography of **Bruce H. Krogh**, please see p. 734 of this TRANSACTIONS.



Medini R. Singh received the B.E. degree in industrial engineering in 1983 from the University of Roorkee, India. He continued with graduate work at the Carnegie Mellon University, Pittsburgh, PA, where he received the M.E. degree in mechanical engineering in 1986, the M.S. degree in manufacturing and operations systems in 1988, and the Ph.D. degree in industrial administration in 1990. This research is part of his graduate work, which dealt with planning and control decisions in computer manufacturing and assembly systems.

He is currently an Assistant Professor of Industrial and Operations Engineering at the University of Michigan, Ann Arbor. His research interests are in general areas of modeling and analysis of manufacturing systems for the development of decision support systems. In particular, through his interaction with IBM, he has been involved in issues like job prioritization and release control in VLSI fabrication lines, production and inventory policies in electronic assembly systems with random yield and component commonality, and coordination between chip production and assembly facilities to improve customer serviceability.

Mr. Singh has been the recipient the IBM Graduate Fellowships for Manufacturing Research, the William Larimer Mellon Fellowship, the Thomas Memorial Gold Medal for engineering design, and many other honors and awards.