Linear Control Rules for Production Control of Semiconductor Fabs

C. Roger Glassey, *Member, IEEE*, Jeyaveerasingam George Shanthikumar, and Sridhar Seshadri

Abstract— **We consider the problem of reducing the cycle time required for producing wafers at a given rate in high-volume single-product semiconductor fabs. Based on theoretical results, we propose a new method of input control that uses intersecting hyperplanes to decide when to release a new lot into the wafer fab. The release control rules constructed thus are said to belong to the class of linear control rules. We provide numerical examples to demonstrate that our method gives nearly optimal results for flowline and probabilistic reentrant flowline models. We then propose the linear control rule called descending control (DEC) and give a hybrid simulation–optimization procedure for determining DEC rules in real-life settings. We provide numerical results for the HP development fab model of Wein [31].**

I. Introduction

In this paper we consider the short- to medium-term production control problem in high-volume single-product semiconductor fabs of reducing the cycle time required for producing wafers at a given rate. The problem of reducing the mean cycle time is important because shorter cycle time can lead to shorter delivery lead time, greater flexibility in meeting customer demands, as well as lower exposure time to contaminants in the clean room, quicker response to yield variations detected by testing of the finished wafer, and lower costs of holding and monitoring unfinished inventories on the shop floor [8], [16], [18], [27], [28], [31], [32]. This problem has been addressed by earlier researchers using analytical as well as simulation techniques. The novel features in this paper are that we use theoretical results from analyzing Markovian models and exact numerical results on optimal control policies obtained by solving a dynamic program to develop a methodology for releasing work into a fab. The methodology for developing work release rules for semiconductor fabs, as proposed by us, combines simulation and optimization techniques in a step-by-step approach.

There are two types of controls available for controlling the production in the fab: i) deciding when and what type of new lot to release into the fab, called the input or release control decision, and ii) deciding which lot will be loaded next at a given work station, denoted the dispatching or scheduling decision. We chose to fix the dispatching rule, and concentrate on the release control decision in this paper. The use of recent advances in scheduling research, described in Section II, in conjunction with our work release method (in a multiproduct setting) is currently under investigation. The method proposed by us was developed in several steps. We initially investigated properties of optimal input control in deterministic as well as failure prone flowlines, and then reentrant flowlines. (An N station flowline is a production system that has either infinite or finite buffers, in which parts after processing at station i, proceed to station $i+1$, eventually exiting the system from the Nth station. In a reentrant flowline, parts make several passes, duplicating the flow from stations $1->2->3-> \cdots ->N$ through the flowline before exiting. In a probabilistic reentrant flowline, parts after finishing processing at station N, with a fixed probability leave the system, else go back to station 1 or 2, and repeat the process.) Next, we studied numerical solutions based on dynamic programming to optimal flow control in flowlines as well as probabilistic reentrant flowlines. We finally combined the ideas from this research with the work of other researchers; and then developed the hybrid optimization–simulation methodology.

At the core of the proposed release control scheme is the state of the shop, which is defined as the vector of jobs under process and awaiting processing at different work centers. Consider the Markovian setting when jobs processing times are exponentially distributed, and we are given a dispatching rule for selecting which job to process next at a work center. Using dynamic programming, it is possible to determine the set of states in which it is optimal to release a new job for processing into the fab. This set is called the optimal control set. In our numerical investigations, the optimal control set was found to be nearly convex but hard to describe using closed form expressions. This finding suggested the approximation of the control set using intersections of linear functions, yielding an approximate control set. The rule suggested by dynamic programming investigations (see Section IV) is to release a new job into the shop whenever the state of the shop lies in the approximate control set. This rule was later called the descending control (DEC) rule because of the ordering of the coefficients in the linear functions. In Section III we describe additional theoretical results that motivated us to consider linear control rules for approximating a control set. Determining the optimal control set for a real life fab is currently impossible, but, as described in Section V, these results (especially the ordering of coefficients), when combined with

Manuscript received October 14, 1993; revised April 16, 1996. The work of C. R. Glassey and S. Seshadri was supported in part by grants to the University of California from the Semiconductor Research Corporation and the California State MICRO Program. The work of J. G. Shanthikumar was supported in part by NSF Grant DDM-9113008 and Sloan Foundation grants for the study on "Competitive Semiconductor Manufacturing."

C. R. Glassey and J. G. Shanthikumar are with the Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, CA 94720 USA.

S. Seshadri is with the Department of Statistics and Operations Research and Operations Management Area, New York University, New York, NY 10012 USA.

Publisher Item Identifier S 0894-6507(96)08120-1.

assumptions about second order properties of throughput, can be used to construct efficient search techniques for determining linear controls. An example of applying the technique to a small flow line and the implementation of such a search technique for the Hewlett–Packard (HP) development fab model (Wein [31]) is described in Section V.

The contributions in this paper are, i) the integration of several theoretical results that bear upon input regulation and scheduling to synthesize a subclass of linear control rules; ii) the use of this class of rules, to provide a unified representation for many input control rules described in literature; iii) the characterization of the optimal control sets for flowlines and reentrant flowlines based on exact solutions to dynamic programs; and iv) a simulation–optimization algorithm that can be used to construct input controls for complex real-life systems, which can not otherwise be analyzed exactly. For the HP development fab model of Wein [31], the DEC rule gave 3%–10% lower average inventory for the same throughput values compared to other methods, such as CONWIP and workload regulation, found in the literature.

In the next section, we examine modeling issues and summarize earlier work relevant to our study. In Section III we provide a summary of theoretical results that motivated the investigation of linear control rules. In Section IV we briefly describe the dynamic programming experiments. In Section V we present the hybrid simulation–optimization extension for determining linear control rules for fabs, and we conclude in Section VI with suggestions for future research.

II. Modeling Issues and Review of Literature

Several books and studies have summarized the manufacturing processes involved in making a VLSI chip (for example see Sze [26] and Sullivan and Fordyce [25]). There are three major stages to making an IC, namely wafer preparation, wafer fabrication, and assembly. We study the wafer fab because most of the complicated production work and also the major investment in facilities are concentrated at the clean room fabrication stage, where the intricate circuitry on the wafer is built up in layers. The work done by several researchers supports the finding that input control, in other words controlling the start-rate of new lots into the fab, is a key variable for controlling throughput and work-in-progress. Significant work done in demonstrating this fact includes that of Glassey and Resende [14] and Wein [31]. Also see Bechte [4] for an early application of input regulation in a different manufacturing context, and Uzsoy, Lee, and Martin-Vega [27] and [28] for a comprehensive review of models that have been used to model semiconductor fabs.

Wein [31] demonstrates the good performance of the workload regulation (WR) rule and Glassey and Resende present the starvation avoidance (SA) rule in [14]. The WR rule was constructed by modeling the fab as a queuing network, and using a Brownian model to approximate the network. When there is only one bottleneck station in the fab, the WR rule is implemented by keeping the total work content for the bottleneck station (as found in all jobs anywhere in the fab) below a preset limit, and new work is released only when the total work content for the single heavily loaded station falls below the preset limit. The starvation avoidance concept is based on the reasoning that: i) the photolithography station is usually a bottleneck in most fabs and also has the most expensive equipment in the fab, and ii) by keeping the workload for this single bottleneck at or below preset limits, the regulation of work in the shop can be achieved. The difference between SA and WR is that Glassey and Resende only consider the work that will arrive at the bottleneck within the time required to feed new work to the bottleneck station, whereas Wein uses the total work content (in jobs anywhere in the fab) for the bottleneck station. Subsequent efforts were made by several researchers to improve the performance of this simple scheme by using sophisticated (but deterministic) queue length predictions. The results were unsatisfactory, because the queue predictions were of poor quality, (for example see Leachman *et al.* [17]). The reader is also referred to Petrakian [20], in which the visit time of jobs to the stations is maintained on a discrete time grid and the production control problem posed in the framework of linear programming. Petrakian's approach can be used for actual shop loading purposes.

Amongst these release control rules, only WR is based on an analysis of the optimal control policy. WR was derived under the assumption of heavy loading and by solving the Brownian model. The analytical extension of WR to fabs with three or more bottleneck stations has proved to be difficult, and interpreting the optimal control policy for the Brownian model to construct a control rule for the physical model has also been found to be a difficult task; for a discussion, see Chevalier and Wein [7] and the references therein. The reader is also referred to a the excellent summary of work done by Kumar *et al.* in [16] and how the intuition derived using queuing as well as other analytical models of the fab could be applied to controlling large real life manufacturing facilities. Other control rules are described in Section III.

The discoveries of new dispatching rules have opened avenues for research into combined input regulation and scheduling techniques. Lu, Ramaswamy, and Kumar [18] show that smoothing the flow of work within the fab can lead to substantial improvements in both the mean as well as the standard deviation of cycle time. Their method employs a "probe" of the downstream stations using iterative simulations to estimate the downstream delay, and thus the slack available for a lot awaiting processing. Similar techniques were also proposed for controlling the release of packets into telecommunication systems; see, for example, Mitra and Seery [19]. Seshadri and Srinivasan [21] and Seshadri and Harche [23] provide a survey and analysis of some scheduling rules used to establish delay bounds in communication networks. These scheduling rules are conjectured to decompose a multiclass queuing network under heavy traffic into separate flowlines, thus making the job of delay estimation and control simpler. Such scheduling techniques have yet to be rigorously combined with input regulation methods.

III. Related Theoretical Results

Our choice of using linear control rules for controlling inputs into the fab and the general modeling approach itself were based on the learning experiences and theoretical results described below. We have chosen to set all the results in one place for easy reference. First, five results are discussed. These provided us the motivation to study the form of optimal control sets using dynamic programming (DP), and the DP formulation is discussed next. In Section IV, we give the description of control rules found in literature and show they are linear and preserve a property called transition monotonicity (TM) described in R1, below.

R1: Consider a k station flowline with reliable machines. Assume that there is infinite demand, unlimited supply of raw materials for the first station, and unlimited buffer (storage) space between all stations. Let the number of jobs under process or awaiting processing at stations i, $i = 2, 3, \cdots, k$ be given by x_i. Let the job processing times be exponentially distributed. Denote $(x_2, x_3, \cdots, x_k)$ as the state of the shop. As an alternate terminology it is appropriate to call this the inventory vector. There may be one or more identical machines at each station. Then, given it is optimal to start processing a new job at the first station in the state $(x_2, x_3, \cdots, x_k)$, we showed in Glassey *et al.* [13] that it is also optimal to start processing a new job in any state that can arise from $(x_2, x_3, \cdots, x_k)$ solely due to the processing of a job(s) at stations 2 to k. Examples being, $(x_2, x_3-2, x_4+1, \cdots, x_k+1)$ and $(x_2, x_3, \cdots, x_k-1)$. This property is called transition monotonicity (TM); see also Glasserman and Yao [11], Veatch and Wein [29], and Weber and Stidham [30]. (In addition we have found that TM is very useful for proving other structural properties of the optimal control policy.) Our linear control rules, as well as other control rules described below, preserve TM.

R2: In Seshadri [22], the shortest expected remaining processing time (SERPT) scheduling rule was shown to be optimal for reentrant flowlines when the discount rate is sufficiently large, and the preemption of jobs is allowed. The near optimality of the SERPT rule when there are many stations or many visits through the flowline was also demonstrated. The SERPT rule has also been found to perform well in empirical investigations as reported in Kumar [15]. As discussed earlier, new scheduling rules outperform SERPT, and also *result* in stable systems for which delay estimation and guarantees can be provided. We have chosen to investigate the input regulation aspect keeping the scheduling rule fixed (to be SERPT in most experiments and first-in first-out (FIFO), in the rest.)

R3: The results reported in Glassey *et al.* [12] convinced us that an input control policy need not completely duplicate the optimal policy to provide good results; also see the conclusions given in Kumar [16]. For example, applying this reasoning the use of an approximate optimal control set, as described in the introduction, need not lead to substantial deviation in performance. A key idea is, however, necessary in creating the approximation (as explained in Section IV), i.e., we must ensure that the weights attached to inventories downstream of the input get progressively smaller in the linear functions.

R4: The work reported in [1]–[3] as well as [13], indicates that failure prone manufacturing facilities can be controlled in either of two ways. If the time to repair is comparable to the time needed to produce a part, then the failure may be treated as a part in process or even as part of the service time (see for example the calculation of the M1–M2 policy shown on p. 383 in [18]). Whereas, if the time to repair was to be very long, say one order of magnitude larger than the processing times, then dropping the input rate into the fab when critical machines fail would be advisable. We have applied our methods first to flowlines and the HP development fab model given in [31] *sans* failures, then heuristically extended the rules to failure-prone fabs without accounting for information on machine failures. The implicit assumption underlying this logic is that the time to repair is relatively short compared to the processing time. Further work is necessary to extend the results reported in [1]–[3] to semiconductor fabs.

R5: The throughput in a single-class closed queuing network with exponential processing times is increasing and concave in the number of jobs in the network, (see Buzacott and Shanthikumar [5]). A similar property can be shown to hold (for the buffer size K) in a M/M/S/K stopped arrival model of a queue, where the input is stopped when the buffer fills up to K. This property of concavity of throughput is conjectured to hold in the threshold level or cutoff value (see Sections IV and V), when the input to a semiconductor fab (producing a single type of part) is controlled using a linear function. The elaboration and application of this property is discussed at the end of Section IV. The property is again used in justifying a quadratic fit in Section V.

Based on these observations, we felt that better control policies than those reported in the literature could be found by: i) using the transition monotonicity property to characterize the states in which it is optimal to release a new job into the system; ii) restricting the search to input control policies by adopting the SERPT dispatching rule; and iii) searching for effective and simple approximations using DP to the optimal control policy while retaining the assumption that processing time distributions are exponential. The dynamic programming setup as well as the use of (and more motivation for using) linear control rules are explained below.

IV. DP Model and Linear Control Rules

In the fab, three major steps are performed per layer, namely oxidation, photolithography, and diffusion, and a typical wafer will undergo these steps 10 times. There are several smaller steps that have to be executed in precise sequence and in a controlled manner within these major steps. The processing times required for carrying out these steps are known, "... within 15 ms to 20 min" depending on the process (p. 51, Sullivan and Fordyce [25]). We reasoned that aggregating these minor operations could prove adequate if the purpose of the analysis is only to gain insights into what may be good operating rules for a fab. The reentrant flowline is the simplest model which can accommodate the repetitive nature of the three major processing steps of oxidation, photolithography, and diffusion, (see for example Uzsoy, Lee, and Martin-Vega

[27], [28]). Kumar [15], [16] has used the reentrant line model to analyze several issues related to production control in semiconductor fabs. However, solving for the optimal control set using DP for even a modest deterministic reentrant flowline model where the product makes five passes through three processing steps proves to be intractable. (The size of the state space, i.e., the total number of permutations of the vector representing the state of the shop, for a small problem could well be of the order of 10^{19}.) Therefore, we took the route of using DP for studying flowlines, then probabilistic reentrant flowlines and finally extending the results heuristically to deterministic reentrant flowlines and semiconductor fabs.

For convenience in solving the DP, we have assumed that there are no yield losses, all machines are available all the time, that lot sizes do not vary from machine to machine, and setup as well as change-over times are negligible. Moreover, we assumed that the processing times of lots are exponentially distributed. We have stated our objective to be the minimization of the cycle time, but for analytical purposes it is easier to consider the equivalent problem of minimizing the average inventory in the fab subject to achieving a given production rate. This problem in turn is equivalent to one of maximizing the average rate of profit over the infinite time horizon, given that each lot on leaving the fab yields a profit of p, and while in the fab incurs a holding cost of c per unit time. Once we had numerical solutions to the DP, given p, c, and the data about the flowlines, we investigated the "shape" of the optimal control set. The logic employed in analysis of the control sets is explained below.

Let the state space be denoted by S. Without loss of generality assume that S is bounded. Define the subset $S(rel)$ by the set of states in which it is optimal to release a job given the profit per unit (p) and a constant holding cost (c) per unit per unit time spent in the system. In symbols, we have $S(rel) = \{X : X \in S\}$ and it is optimal to release a job when the state is X. This set, $S(rel)$, is the *optimal control set*. Define a *partial order* based on transition monotonicity (TM), $X \geq_{\text{TM}} Y$, if the state Y can be reached from X without input of new jobs into the system. The order induced by TM is a partial order because in general given two nonnegative n-dimensional real vectors, we may not be able to compare them (i.e., say which one is smaller and which is larger) using TM. For example, the vectors, $(2, 1, 3)^T$ and $(3, 0, 2)^T$ cannot be compared using TM. However, we may use this ordering to further reduce the control set by defining

$$S(rel \geq_{\text{TM}}) = \{X : X \in S(rel \geq_{\text{TM}}); \text{ and } Y \in S(rel \geq_{\text{TM}}), \; Y \leq_{\text{TM}} (\geq_{\text{TM}}) X \Rightarrow Y = X\}.$$

This is called the reduced control set and contains all the largest elements of the chains ordered by TM. This set is possibly smaller than $S(rel)$, but still it is impossible to enumerate the set without solving a dynamic program. Instead consider the set of all linear functions f such that $X \geq_{\text{TM}} Y \Rightarrow f(X) \geq f(Y)$. Define the cutoff value c_f for f as the $\max\{f(X) : X \text{ belongs to } S(rel)\}$. Then the set $S(rel)$ is surely a subset of $S_f(c_f) = \{X : X \text{ belongs to } S \text{ and } f(X) \leq c_f\}$. The class of input control rules based on the above reasoning will be called the class of *linear control rules that preserve TM*. The class of rules that use combinations of linear functions to effect control will be called the class of linear rules. If we use intersections and unions of the $S_f(c_f)$'s, then the control set can at least in theory be described very accurately. For example we could choose approximations of the form $S_{app} = S_f(c_f) \cap S_g(c_g) \cup S_h(c_h)$. (The control sets were found to be convex in our numerical examples. Therefore intersections of the hyperplanes, $S_f(c_f)$'s, should be adequate for approximating the control set, as explained at the end of this section.) This was the basic motivation for performing the set of experiments described next.

We first investigated four-stage flowlines, with multiple but reliable machines at each stage. The job processing times are assumed to be exponentially distributed. The data for a problem (see Table I) consisted of i) four pairs of numbers denoting the number of machines and the processing rate of a typical machine for each of the four stations; ii) values of profit per unit produced (p) and holding cost per unit time per job (c); and iii) upper bound on total inventory allowed in the system. The average cost criterion was used for solving the DP to maximize the profit rate. In order to modify the above set up to model reentrant flowlines, we permitted *probabilistic reentrance* back to the second station of the flowline. In this model, a job on completion of processing at the fourth station is fed back to the second station with a fixed and given probability. This device allows us to work with a manageable state space but does not permit modeling of different service times for different visits to the same station. Several rules were used as benchmarks and also as functions for approximating the optimal control set. The rules described below are of interest because they are widely found in the literature *and* they are linear and preserve TM. A brief description of the rules, and their application to a four-stage flowline is given. In the examples, the number of jobs at station i, $i = 2, 3, 4$, is denoted as x_i, $i = 2, 3, 4$. See Section II for a description of the first two rules.

1) Starvation Avoidance (SA): For a four-station flowline, if the third station were the bottleneck station, then the SA rule is to release new jobs if $(x_2 + x_3)$ is below a preset limit. The rule is linear and seen to preserve TM.

2) Workload Regulation (WR): When there is a single bottleneck, this rule is the same as SA. We have assumed that any station which is loaded 90% or less relative to the bottleneck station is a nonbottleneck. With this qualification, in most cases for flowlines, SA and WR are identical. WR also belongs to the class of linear control rules.

3) CONWIP: The idea in this rule is to keep the total number of jobs in the system below a certain level (see [8]–[10] and [24]). This rule is also a linear control rule, preserves TM, and for the four-station example we release a job if $(x_2 + x_3 + x_4)$ is below a preset limit.

4) Deterministic (DET): In this rule, work is fed into the network at constant intervals of time depending on the throughput required from the network. Implementing this rule leads to open loop control and is included for comparison purposes.

TABLE I
DATA FOR FOUR-STAGE FLOWLINE AND PROBABILISTIC REENTRANT LINE MODELS

	Station 1		Station 2		Station 3		Station 4		
Model	No. of Machine	Process. Rate per Machine (Rel..Load)*	No. of Machine	Process. Rate per Machine (Rel..Load)*	No. of Machine	Process. Rate per Machine (Rel..Load)*	No. of Machine	Process. Rate per Machine (Rel..Load)*	Probability of Reentering 2nd Stn.**
FLOW1	2	0.8 (75)	4	0.3 (100)	3	0.45 (89)	4	0.375 (80)	0
FLOW2	2	0.8 (75)	4	0.3 (100)	3	0.42 (95)	4	0.325 (92)	0
FLOW3	2	0.8 (84)	4	0.5 (67)	3	0.45 (99)	4	0.335 (100)	0
FLOW4	2	0.8 (75))	4	0.3 (100)	3	0.42 (95)	4	0.305 (98)	0
RFLOW1	2	0.8 (25)	4	0.3 (100)	3	0.42 (95)	4	0.305 (98)	0.6666
RFLOW2	2	0.8 (15)	4	0.3 (100)	3	0.45 (89)	4	0.375 (80)	0.8

Notes: * - Relative load is computed with respect to bottleneck station.
**- Probability that job on leaving last staion reenters the 2nd station.

5) The PAC System: The production authorization card (PAC) system was developed by Buzacott and Shanthikumar [5], [6]. When PAC is applied in a simplified form to a four-station flowline, we obtain $S(rel) = \{x_2, x_3, x_4 : \min(Z_3 - x_2 - x_3 - x_4, K_3 - x_2 - x_3, K_2 - x_2, K_1) > 0\}$, where Z_3, K_i, $i = 2, 3, 4$ are parameters. Thus PAC also belongs to the class of linear control rules, and can be shown to preserve TM. Moreover PAC can mimic the control exercised by Kanban, MRP, base stock policy, CONWIP, and fixed buffer.

6) Function TWK: This function measures the total work content of the jobs in the system. Let the combined processing rates of the machines at the three downstream stations be (μ_2, μ_3, μ_4). Let the inventory (including the jobs under process) at the three downstream stations be $X = (x_2, x_3, x_4)$. Then

$$TWK[(\mu_2, \mu_3, \mu_4), X] = \frac{x_2}{\mu_2} + \frac{x_2 + x_3}{\mu_3} + \frac{x_2 + x_3 + x_4}{\mu_4}$$

TWK preserves TM.

7) Function NJOB: This function adds up the number of jobs in the system at the downstream stations, $NJOB(X) = x_2 + x_3 + x_4$.

8) Function BOT: This function adds up the jobs at the bottleneck station. For example if station 3 is the bottleneck, then $BOT(X) = x_3$ BOT does not preserve TM.

The usefulness of the compact representations of the control set given above is that now all the rules can be exercised by using linear controls. A simulator needs only to know how many linear functions are there, what the coefficients are to attach to the various components of the current inventory vector and an associated cutoff (or threshold) value for each function. The controller then chooses to release a job if the first station has an available machine and if all the functions evaluated at the current value of the inventory vector return numbers below the corresponding threshold or cutoff values. The result R5 described in Section III, can be used to fine-tune the value of the threshold. As an example, consider the rule, $3x_2 + 2x_3 + x_4 \leq k$. If we get throughput values of TH_1 and TH_2 using values of $k = k_1$ and k_2, using R5, we may use linear interpolation to guess the correct value of k for obtaining throughput values between TH_1 and TH_2, or use bisection search to find the right value of k for a desired throughput rate. This trial and error device is necessary, whichever rule is being considered, SA, WR, or a general linear control.

The models FLOW1–FLOW4 were used for testing the use of linear control rules in flowlines. The data for these models is given in Table I. For each model, given the values for the profit, p, and the holding cost, c, we first obtained the optimal control set by solving a dynamic program. Then we tried out various combinations of *just* the rules described in the previous section (linear functions) to determine which pair of functions gave the *best* fit to the optimal control set (see Table II for some examples). The fit was obtained by eyeballing the data and the details can be found in Seshadri [22]. The approximate control set was then tested for efficacy using simulation and the results are shown in Table II. As seen from Table II, the intersecting hyperplane rule performed as well or better than other rules compared in the experiments. The best benchmark in these simulations is against the PAC rule, because as mentioned earlier, PAC can mimic many of the linear control rules found in the literature. The results corresponding to CONWIP are not shown, but the rule gave poor performance. In the second stage of the experiments, we used reentrant flowline models

TABLE II
SOME SIMULATION RESULTS FOR FOUR-STAGE FLOWLINE MODELS

Model	Utilizatio of Bottle -neck	Best Fit Control			Starvation Avoidance		PAC System		WR	
		Pair of Functions Used	Throughput Rate	Average Inventory x 0.3	Throughput Rate	Average Inventory x 0.3	Throughput Rate	Average Inventory x 0.3	Throughput Rate	Average Inventory x 0.3
FLOW1 (*)	98.40%	TWK+BOT	1.181 +/- 0.046	6.41 +/- 0.06	1.181 +/- 0.046	6.42 +/- 0.11	Same as SA		Same as SA	
FLOW2 (*)	99.50%	TWK+BOT	1.194 +/- 0.004	11.08 +/- 0.37	1.194 +/- 0.004	11.22 +/- 0.64	Same as SA		Same as SA	
FLOW3 (*)	97.70%	TWK+NJOB	1.309 +/- 0.004	10.17 +/- 0.06	1.308 +/- 0.004	11.34 +/- 0.04	1.309 +/- 0.016	10.66 +/- 0.28	Same as SA	
FLOW4 (*)	98.90%	TWK+BOT	1.187 +/- 0.005	13.99 +/- 0.55	1.188 +/- 0.006	16.53 +/- 1.97	1.188 +/- 0.004	14.45 +/- 0.36	Same as SA	
RFLOW1 (*)	97.70%	NJOB+BOT	0.3908 +/- 0.005	11.212 +/- 0.312	0.3905 +/- 0.005	11.911 +/- 0.312	Same as Best Fit		0.3898 +/- 0.005	12.445 +/- 0.172
RFLOW2 (*)	99.60%	TWK+BOT	0.23899 +/- 0.004	7.347 +/- 0.283	0.23818 +/- 0.004	7.061 +/- 0.408	0.23818 +/- 0.004	6.947 +/- 0.188	0.23884 +/- 0.004	10.346 +/- 0.48

Note: (*) 99% Confidence intervals, Simulation run length = 100,000 to 500,000 time units
Confidence intervals based on breaking up observations for blocks of 5,000 units

with probabilistic routing. Table I gives the data for the models and Table II the simulation results using the best pair of functions from the list. These six experiments convinced us of the following. i) Using intersecting hyperplanes for control purposes gave as good or better performance compared to other rules found in the literature. ii) Rules such as SA, WR, and PAC are not robust in the sense we can come up with counter examples wherein such rules fail. iii) The intersecting hyperplane rule is better because only its functional form is specified and there is flexibility in choosing the coefficients based on experimentation.

These observations alone would not suffice to develop good linear controls for fabs. However, we draw attention to three important points about a typical control set. The first observation is that the control sets are nearly convex. Second, it follows that *intersection* of linear functions would suffice to approximate the control set, and the numerical results suggest that the approximation did not detract much from the quality of the solution. Third, the slopes of the set indicate that it is important to place more weight on upstream inventories. To help visualize the first and last observations, we have provided sections of the optimal control set obtained from solving a DP, for a four-station balanced flowline, in Fig. 1(a)–(e). The sections (plots of the control set in the x_2 and x_4 plane) are taken along the x_3 plane. The optimal control, the "best fit linear control," and the SA control are shown. Note that SA remains a "vertical line" throughout these sections, whereas the approximating hyperplanes adjust to increasing congestion at the third station. We used these ideas in developing linear controls for models from the class of *deterministic reentrant flowlines* and the details can be found in [22]. In these experiments, we were able to obtain 5%–40% improvement in cycle time over SA or WR, using linear controls constructed by trial and error. The important point to be kept in mind in developing the control was to place more weight on upstream inventory. Thus if the product undergoes n steps of processing in the fab, then letting x_i, $i = 1, 2, \cdots, n$, be the number of jobs at step i, the form of the linear function should be $a_1x_1 + a_2x_2 + \cdots + a_nx_n$, with $a_i \geq a_{i+1}$, $i = 1, 2, \cdots, n-1$. Such a (descending) linear function preserves TM if the route through the shop were visualized to be a "long" flowline. In the next section we discuss how a combined simulation–optimization approach can be used to develop such a function in a step-by-step manner.

V. HYBRID SIMULATION-OPTIMIZATION APPROACH FOR CONSTRUCTING LINEAR CONTROLS AND NUMERICAL RESULTS

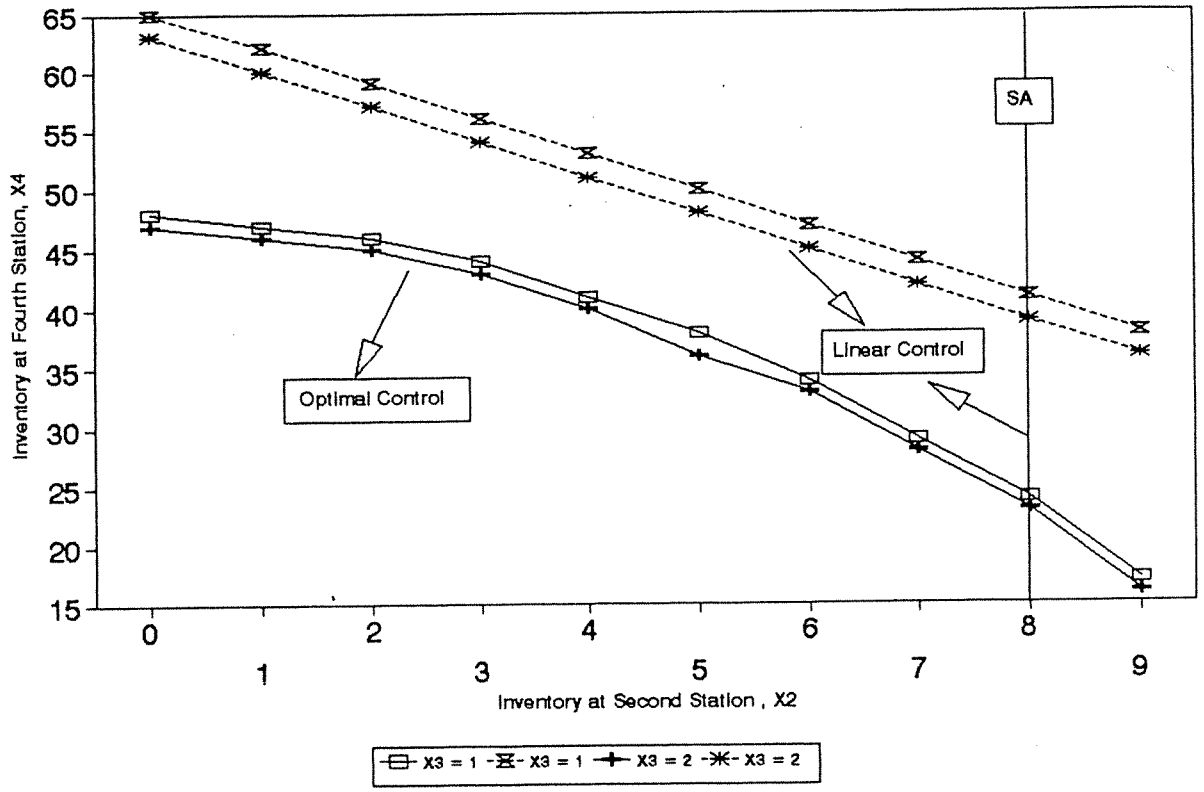
We present a simulation–optimization approach for developing linear control rules in this section, and discuss some results. An algorithm such as the one presented below is not new, and similar algorithms have been used to carry out sensitivity analysis in different contexts (see for example the discussion of stochastic optimization in Glasserman and Yao [11]). We propose the following numerical procedure for obtaining the coefficients, a_i's, in $a_1x_1 + a_2x_2 + \cdots + a_nx_n$, without solving a dynamic program.

A. Simulation–Optimization Procedure

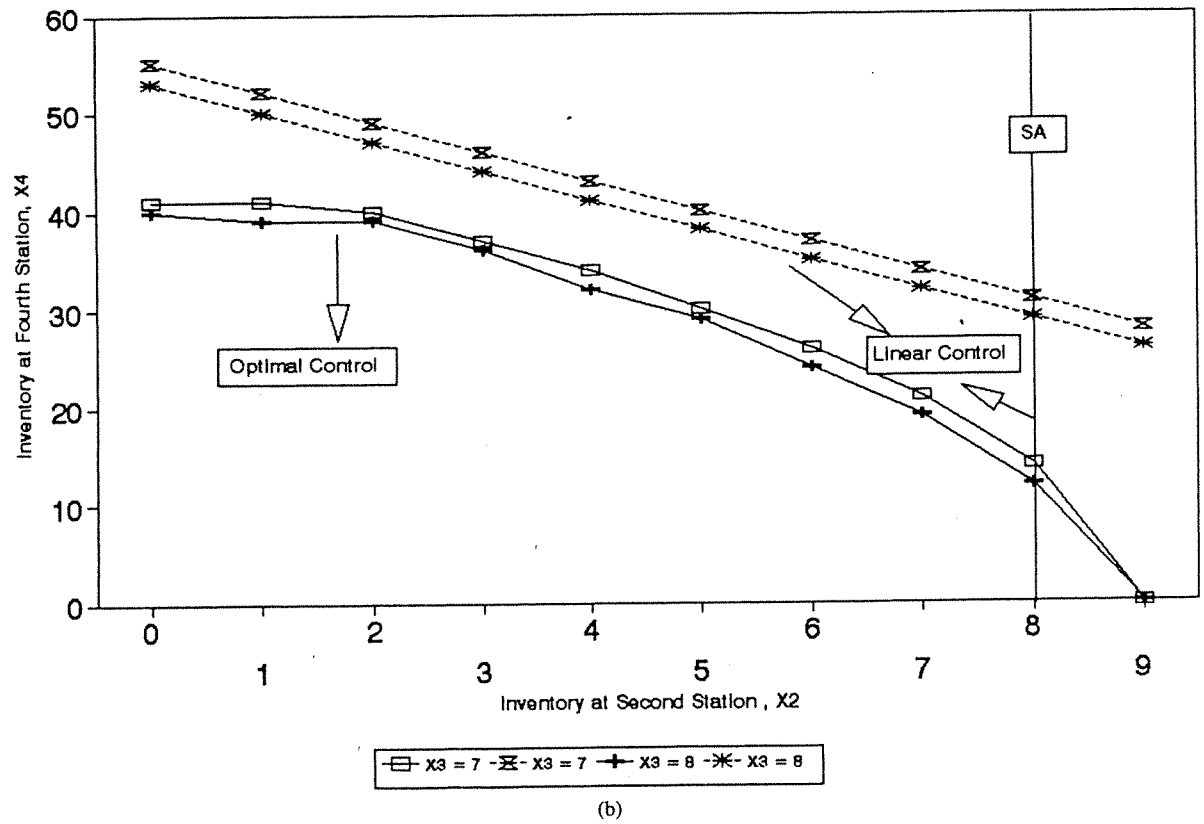
0. Input = desired value of throughput, TH.
Set $a_0 = x_0 == 0$. x_i is the number of jobs at step i of the route through a fab. a_1 is fixed to be 1 throughout. n = number of steps in the process (route).

1. **DO** $i = 2, \cdots, n$

1.1 (Simulation) Using the hyperplane $a_1x_1 + \cdots + a_{(i-1)}x_{(i-1)} + \alpha x_i \leq b$, keep varying b till TH is



(a)

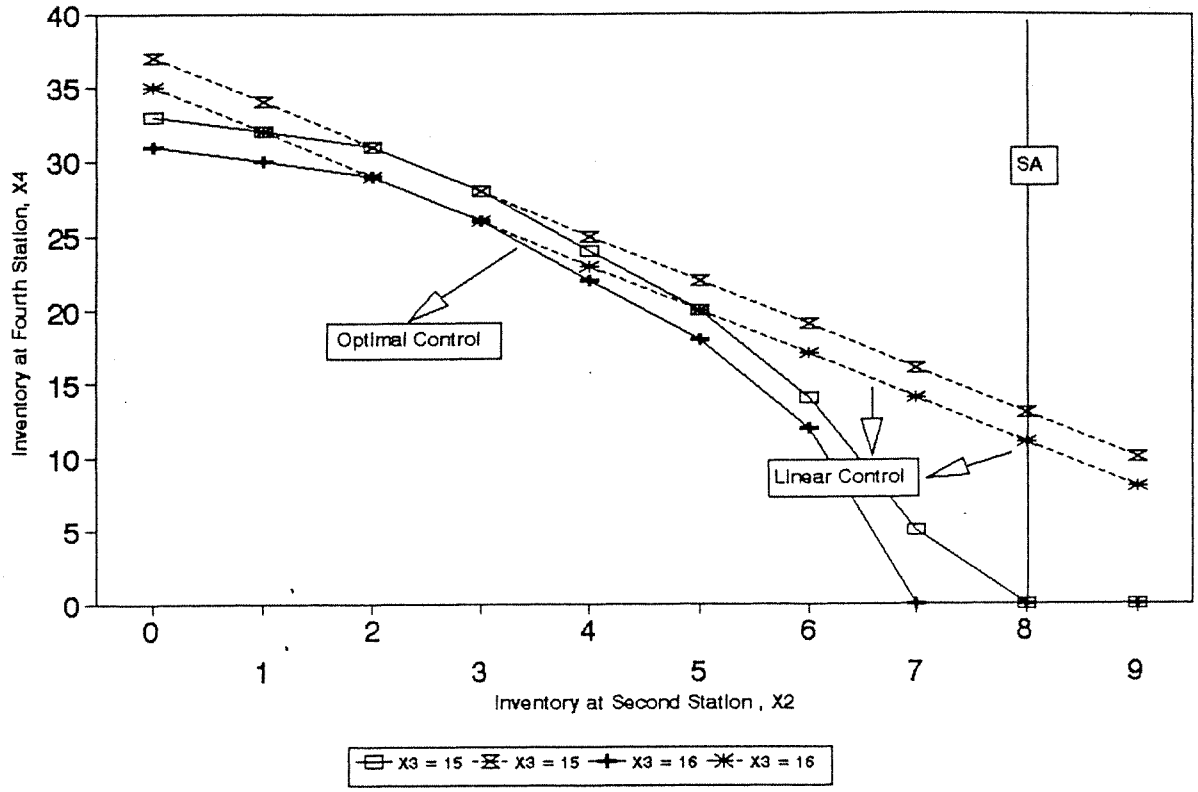


(b)

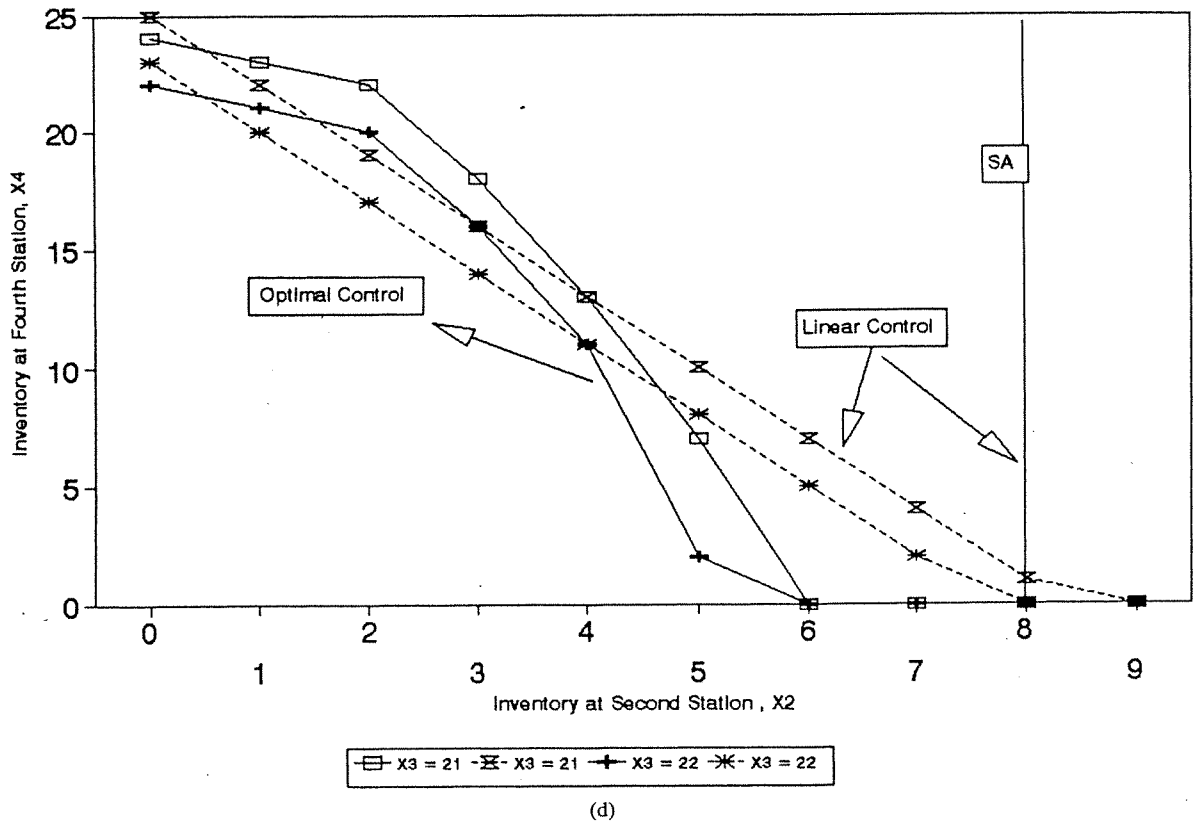
Fig. 1. (a) Balanced flowline: section at X3 (3rd stn inv) = 1 and 2 and (b) balanced flowline: section at X3 (3rd stn inv) = 7 and 8.

attained, for values of $\alpha = 0$, $a_{\{i-1\}}/4$, $a_{\{i-1\}}/2$, $a_{\{i-1\}}$. (In practice the trial values must be chosen such that at least one value of α gives a lower total average inventory compared to $\alpha = 0$.)

1.2 Label the total average inventory values, as $I(\alpha)$, $\alpha = 0$, $a_{\{1-1\}}/4$, a_{\{i - 1\}}/2, $a_{\{i-1\}}$.



(c)

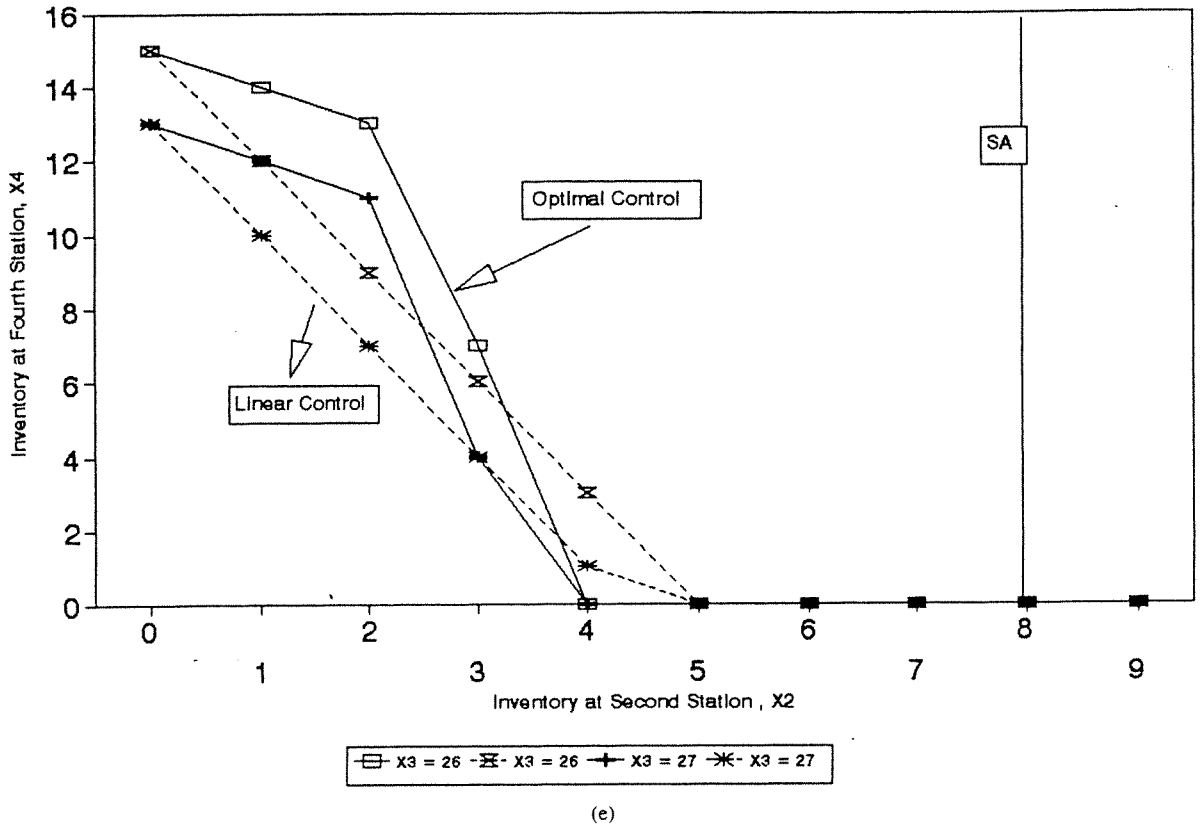


(d)

Fig. 1. (*Continued.*) (c) Balanced flowline: section at X3 (3rd stn inv) = 15 and 16 and (d) balanced flowline: section at X3 (3rd stn inv) = 21 and 22.

1.3 Fashion a quadratic fit, $I(x) = a + bx + cx^2$; using the data in step 1.1. If no trial value gives average inventory which is smaller than for $\alpha = 0$, go to next i after setting $a_i = 0$.

1.4 (Optimization) Minimize the quadratic function obtained in step 1.3.



(e)

Fig. 1. (*Continued.*) (e) Balanced flowline: section at X3 (3rd stn inv) = 26 and 27.

1.5 Simulate and test the new hyperplane, repeat step 1.1 using the additional information if necessary.

ENDDO

2. STOP

Remarks:

1) In step 1.1, some simulation is necessary to obtain the correct threshold value corresponding to the throughput TH, and typically3–4runs are necessary. As mentioned earlier we rely on result R5, and the fact that any linear control rule needs some trial and error for obtaining the desired value of throughput.
2) In step 1.3, we assume that the effect of the ith coefficient can be captured using a quadratic function. Typically, the term b in the quadratic, $I(x) = a + bx + cx^2$, will be negative and$c$ will be positive. This would follow if R5 were true in this setting, because $-I(x)$ would then be concave in x. Another justification for using a quadratic function is the idea of a Taylor series expansion around $x = 0$.
3) Sometimes, it is difficult to get the exact value of TH for all trial values of α in step 1.1. In that case, the resulting average inventory can be normalized by first multiplying by $(1 - \rho_{bot})$, where ρ_{bot} is the utilization of the bottleneck station. This procedure is justified by recourse to the formula for the average number of customers in queue in a M/G/1 queuing system, and in general by the form of delay bounds, such as those given in Buzacott and Shanthikumar [5] or Seshadri and Srinivasan [21]. It is also difficult to obtain the value of TH, using just say control of inventory at the first two steps of processing. In such situations, we suggest the use of CONWIP (in addition to the hyperplane being developed) to control the throughput to the desired level (see Table V).
4) In a practical setting, such as a semiconductor fab, the number of steps, n, can be very large. For example, in the HP development fab model [31] there are 172 processing steps using 24 stations. In such cases, executing the DO loop could be time consuming. However, based on the logic behind WR as well as SA, we may aggregate the inventory between visits to highly loaded stations. For example, if there were two bottleneck stations, inventory between visits to either station could be aggregated (see Tables III–V).
5) As a thumb rule, we need 3–4 simulations per trial value per visit to a highly loaded station. Thus a total of say (4 simulations × 4 trial values × the number of visits to a highly loaded station, say m) $=16m$ simulations. This number can be very large, but the entire algorithm should then be placed on auto-pilot, and the method coded to work (off-line) without manual intervention. For an exhaustive analysis and search on a standalone workstation, the exercise could well take 10 min per simulation, and up to 8 h of work for $m = 30$. Our experience running the simulations indicates that the time required will be smaller, as discussed in the next section.

TABLE III
ONE-BOTTLENECK EXAMPLE

Input Rule	Throughput Rate	Average Inventory*0.3	Improvement (%) over Deterministic	Comment
Deterministic	0.03053	7.9820		95-96% utilization of bottleneck
CONWIP	0.03048	8.0470	-0.81	
WR	0.03049	7.9690	0.16	
Linear Control	0.03057	7.8380	1.80	Steps 1-4 weight = 3.15 Steps 5-13 weight = 1 Steps 42-62 weight = 0.693
Deterministic	0.03127	9.0990		97-98% utilization of bottleneck
CONWIP	0.03129	9.2616	-1.79	
WR	0.03129	9.0177	0.89	
Linear Control	0.03131	8.8426	2.82	Steps 1-4 weight = 1 Steps 5-13 weight = 0.837
Deterministic	0.03181	11.7213		99.5% utilization of bottleneck
CONWIP	0.03183	11.6500	0.61	
WR	0.03182	11.2191	4.28	
Linear Control	0.03182	10.4236	11.07	Steps 1-4 weight = 1 This corresponds to SA

Notes: This example corresponds to Wein [31]'s one bottleneck model. All machines are failure free.
Processing time distribution is exponential, scheduling rule SERPT, simulation length = 1,000,000.
Standard error of estimates based on blocks of 5,000 time units data was typically of the order of magnetude of 1.0E-04 for throughput rate, and 1.0E-02 for the average inventory.
Bottleneck station is PHOTO.

6) Finally, we point out the importance of the "form" of the linear control. Because the coefficients are descending, the search in step 1.1 can be restricted to just a few values. This provides the efficiency of the method. The case for using descending coefficients can be further strengthened if the new scheduling rules such as the fair processor sharing rule described in Seshadri and Srinivasan [21] are used. Under the use of such rules that *mimic* head-of-the-line processor sharing, the reentrant line behaves very much like a long flow line. Such a rule would also simultaneously achieve the objective of smoothing the internal flow (cf., [18]) if the processing times were deterministic.

We give below an example of applying a variation of the method to create a linear control for a small flowline, and then discuss results obtained for the HP development fab model. Consider the four-station flow line with 2, 4, 3, and 4 identical machines at the four stations. Let the processing times be exponentially distributed (we also tested with gamma distributions having different shape parameters), with mean processing times 1.25, 2, 2.22, and 2.985 at the four stations. The desired value of throughput, TH, is 1.31. This corresponds to 96.9% utilization of the bottleneck station (i.e., station #4). As in Section IV, in all results the average inventory has been multiplied by 0.3 (as the simulator uses a holding cost per unit time of 0.3). We have cut and pasted simulation outputs, to avoid errors in transcribing results. Using the hyperplane, $x_3 + x_4 \leq 55$ we get TH of 1.31000 and average inventory of 18.2877. Using the hyperplane, $x_3 \leq 13$, we obtained a TH of 1.31047 and average inventory to be 20.5643. Using the hyperplane, $x_4 \leq 48$, the values of TH and average inventory are 1.31203 and 87.3755. The system is "uncontrollable" using the hyperplane $x_2 \leq b$ and so we ignore the inventory at station 2. Using the above results, let x be the *relative* weight given to x_4 [i.e., the ratio of $a_4/(a_2 + a_3 + a_4)$]. Then we need to fit the quadratic, $I(x) = a + bx + cx^2$ using

$$\begin{aligned} a &= I(0) \\ &= 20.56 \\ a + b(0.5) + c(0.5)^2 &= I(0.5) \\ &= 18.28 \\ a + b + c &= I(1) \\ &= 87.37. \end{aligned}$$

Solving for a, b, and c, we get $I(x) = 20.56 - 75.9x + 142.92x^2$. Minimizing in x, we get $x = 0.266$, and the desired new hyperplane as $2.76x_3 + x_4 \leq b$, [where we used the result from the optimization $x = 1/(1 + 2.76) = 0.266 =$

TABLE IV
TWO-BOTTLENECK EXAMPLE

Input Rule	Throughput Rate	Average Inventory*0.3	Improvement (%) over Deterministic	Comment
Deterministic	0.03043	11.8906		95.2% utilization of bottleneck
CONWIP	0.03049	12.1791	-2.43	
WR	0.03046	11.8732	0.15	
Linear Control	0.03047	11.3111	4.87	Steps 1-4 weight = 1 Steps 5-13 weight = 0.87
Deterministic	0.03114	15.2758		97.2% utilization of bottleneck
CONWIP	0.03105	15.1450	0.86	
WR	0.03110	15.0464	1.50	
Linear Control	0.03114	13.8624	9.25	Steps 1-4 weight = 1 Steps 5-13 & step 17 weight = 0.05
Deterministic	0.03147	19.8152		98.4% utilization of bottleneck
CONWIP	0.03145	19.0504	3.86	
WR	0.03151 & 0.031441	18.8756 18.1204	4.74 8.55	
Linear Control	0.03150	17.1866	13.27	Steps 1-4 weight = 1 This corresponds to SA
WR	0.031568	13.9657		Changed processing time distribution to gamma parameter = 2 98.7% utilization of bottleneck
Linear Control	0.031568	12.5020	10.48 (over WR)	Steps 1-4 weight = 1 Steps 5-13 & 17 weight = 0.031

Notes: This example corresponds to Wein [31]'s two bottleneck model. All machines are failure free.
Processing time distribution is exponential, scheduling rule SERPT, simulation length = 1,000,000.
Standard error of estimates based on blocks of 5,000 time units data was typically of the order of magnetude of 1.0E-04 for throughput rate, and 1.0E-02 for the average inventory.
Bottleneck stations are PHOTO (1st two visits = steps 4, 13) & ION IMPLANT (first visit = step 17).

$75.9/(2 * 142.92)]$. The hyperplane suggested is quite unlike workload regulation, starvation avoidance or CONWIP. This hyperplane when used to control the flowline gives, TH = 1.31121, and average inventory to be 15.1264. Note the 20% improvement (improvement = 18.28/15.12) over giving equal weight to x_3 and x_4 (i.e., $x_3 + x_4 \leq 55$) as suggested by the workload regulation method. At higher values of TH we obtained larger percentage improvement over WR or SA.

We next discuss examples using Wein's [31] model of a fab. The fab has 24 stations and 172 processing steps. We assume that all stations are free from failures. Most of our simulation examples assume exponential processing times and the SERPT scheduling rule. Some results are shown for failure-prone stations, FIFO scheduling as well as processing times with gamma distribution of parameter = 2 (which is Erlang of degree 2). In these examples, station #14 (LITHOGRAPHY) is one of the bottleneck stations. We give examples with one-, two-, and four-bottleneck stations. In these examples, jobs make the first visit to station #14 on their 4th step and the second on their 13th step. The actual data is in the public domain, and the interested reader is referred to Lu, Ramaswamy, and Kumar [18], and Wein [31]. Due to limitations on our computing resources, we have developed only one linear function per example, and stopped the algorithm when reasonable improvement in cycle time was obtained compared to either WR or Deterministic release. The simulator employed was coded in *f77* for reentrant flow models and adapted to cater to linear control rules. The processing time distributions were truncated at seven times their mean. As mentioned in the remarks at the beginning of this section, for computing the linear control, we had to "normalize" the average inventory because the throughput values will not be exactly the same using different linear controls.

For one- or two-bottleneck fabs (see Tables III and IV), we are able to obtain 3%–10% reduction in the average inventory, with greater reduction being observed for more heavily loaded fabs. We have labeled one of the release control rules as WR in Table IV, but follow the surrogate release control policy suggested in the appendix of Wein [31], p. 129. The last example in Table IV corresponds to processing

TABLE V
FOUR-BOTTLENECK EXAMPLE

Input Rule	Throughput Rate	Average Inventory*0.3	Improvement (%) over Deterministic	Comment (no WR rule known for this example)
Deterministic	0.02914	10.0721		91% utilization of bottleneck SERPT scheduling
CONWIP	0.02901 & 0.02920	9.8285 10.1214	2.42 -0.49	 SERPT
Linear Control	0.02916	9.8079	2.62	Steps 1-4 weight = 1 Steps 5-13 weight = 0.56 CONWIP to control, SERPT
Deterministic	0.02998	11.7646		93.8% utilization of bottleneck SERPT scheduling
CONWIP	0.03001	12.0394	-2.34	
Linear Control	0.02944	11.4424	2.74	Steps 1-4 weight = 1 Steps 5-13 weight = 0.1 CONWIP to control, SERPT
Deterministic	0.02999	9.8242		93.8% utilization of bottleneck FIFO scheduling
Linear Control	0.02999	9.6388	1.89	Steps 1-4 weight = 2 Steps 5-41 weight = 1, rest weight = 0.1 FIFO scheduling
Deterministic	0.03107	13.6080		97.2% utilization of bottleneck FIFO scheduling
Linear Control	0.03105	13.1147	3.63	Steps 1-4 weight = 2 Steps 5-41 weight = 1, rest weight = 0.1 FIFO scheduling
Deterministic	0.03144	18.0872		98.3% utilization of bottleneck FIFO scheduling
Linear Control	0.03149	16.7716	7.27	Steps 1-4 weight = 2 Steps 5-41 weight = 1, rest weight = 0.1 FIFO scheduling
Deterministic	0.02982	16.1638		94.2% utilization of bottleneck FIFO scheduling Stations 13-24 failure prone (see below)
Linear Control	0.03001	14.9549	8.08	Steps 1-4 weight = 2 Steps 5-41 weight = 1, rest weight = 0.1 FIFO scheduling

Notes: This example corresponds to Wein [31]'s model with stations 14 19, 21 & 24 being bottlenecks.
The number of machines at these 4 stations are 3,1,1,&1 respectively.
Processing time distribution is gamma parameter =2, scheduling rule varies, simulation length = 1,000,000.
Standard error of estimates based on blocks of 5,000 time units data was typically of the order of magnetude of 1.0E-04 for throughput rate, and 1.0E-02 for the average inventory.
In the last example, MTTF = 10,000 and MTTR = 100, with failure & repair times exponentially distributed, for stations 13 to 24 (that includes all bottleneck stations).

time distribution of gamma with parameter equal to 2. For the four-bottleneck fab model, see Table V, FIFO scheduling is better (as already noted in classical job shop scheduling literature). There is no known WR or surrogate WR method for this example. For this four-bottleneck example, processing time distributions are gamma of parameter equal to 2; and we see that changing the service time distribution does not change the relative magnitude of improvement in cycle time. We also show one example (the last one) in Table V, where all the four-bottleneck stations have mean time to failure (MTTF) and mean time to repair (MTTR) of 10 000 and 100 respectively. Both the time to fail as well as repair times are exponentially distributed. In general we observe that for failure-prone fabs the improvement due to controlled release over deterministic release occurs earlier, especially when the mean time to repair is larger than the processing time. (Initial experiments with fab models, in which the MTTR is the same order of magnitude as the service times, indicate that a modified algorithm might be necessary. When determining the ith value of the coefficient, the linear function should be of the form $a_1x_1 + a_2x_2 + \cdots + a_{(i-1)}x_i + a\sum_{j\geq i} x_j$. The value of "$a$" is selected as done before by optimizing the quadratic function fitted using simulated values. Improvements of 3%–10% in the average inventory are observed for the single-bottleneck examples.)

In all the examples, at very high levels of bottleneck utilization, the coefficients of the linear control are zero beyond the initial few steps. This suggests that input regulation helps in reducing congestion at the very beginning of the process, and the rest of the inventory is hard to control. Obviously, if this

were true, then the search for coefficients in the algorithm can terminate sooner, and this issue needs further experimentation and research. One way of overcoming the lack of control beyond the initial steps is to stage the inventory every so many steps, and release from these stages in a controlled manner. The dispatching algorithm given in [18] apparently achieves this end.

VI. Conclusions

We have demonstrated a practical algorithm that can be used to develop linear control rules for input regulation in semiconductor fabs. The class of linear control rules have been narrowed down into those with descending coefficients. This narrowing down helps in searching for a good linear control, as explained in the remarks following the simulation–optimization procedure. Several interesting theoretical issues arise from these investigations. First, there is the role of the scheduling discipline. For fabs that have several bottleneck stations, it appears that FIFO is still a good rule to adopt. There are other dispatching rules that have not yet been tried out in fabs, such as fair processor sharing. These rules have been used to guarantee delay in telecommunications networks, and recently Seshadri and Harche [23] established bounds for delay in multiclass open queuing networks using such a rule. The interesting feature about these rules is that under heavy traffic and deterministic routing, the network is conjectured to decompose into separate tandem queues. These individual tandem systems are more amenable to analysis, and can be controlled as if they are single product systems. Such rules are also more natural in a multiproduct setting, and we intend to test these rules in a multiproduct setting in future work. Second, in our experiments most of the improvement in performance, due to controlled release, comes at the early processing steps. Therefore, there is still scope for applying dynamic programming to the initial steps to compute an approximate release control policy. Establishing the validity of this procedure in terms of bounds for performance is an interesting problem. We have also suggested that staging inventory and controlled release from WIP may be another alternative worth exploring. Third, the use of hierarchical procedures for controlling failure prone fabs—even a simulation analysis that uses "discrete" flow of jobs (as against continuous flow approximation) is a possible extension of this work. Fourth, given a sequence of jobs that have to be released into the fab and a scheduling rule that uses due dates (such as the one described in [18]), the linear control rules described in this paper can be adapted to make-to-order situations. Fifth, iteratively determining the control, observing the results, and making changes based on the values of the observed deviations in average inventory at different steps, is a feature that can be combined with the simulation–optimization approach to develop adaptive control mechanisms.

Acknowledgment

The authors would like to thank the Associate Editor and two anonymous referees for their suggestions and comments. Their suggestions that a practical method should be given for developing linear control rules led to the simulation–optimization algorithm described in Section V.

References

[1] R. Akella, Y. Choong, and S. B. Gershwin, "Performance of hierarchical production scheduling policy," *IEEE Trans. Comp., Hybrids, Manufact. Technol.*, vol. 7, pp. 225–240, 1984.

[2] R. Akella and P. R. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *IEEE Trans. Automat. Contr.*, vol. 31, pp. 116–126, 1986.

[3] R. Akella, S. Rajagopalan, and M. R. Singh, "Part dispatch in random yield multistage flexible test systems for printed circuit boards," *Oper. Res.*, vol. 40, pp. 776–789, 1992.

[4] W. Bechte, "Theory and practice of load oriented manufacturing control," *Int. J. Prod. Res.*, vol. 26, pp. 375–395, 1988.

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

[6] ——, "PAC: A general approach for coordinating production in multiple-cell manufacturing systems," *Prod. Oper. Manage.*, vol. 1, pp. 34–52, 1992.

[7] P. B. Chevalier and L. M. Wein, "Scheduling networks of queues: Heavy traffic analysis of a multistation closed network," *Oper. Res.*, vol. 41, pp. 743–758, 1993.

[8] I. Duenyas, J. W. Fowler, and L. W. Schruben, "Planning and scheduling in Japanese semiconductor manufacturing," *J. Manuf. Syst.*, vol. 13, pp. 323–332, 1995.

[9] I. Duenyas and M. F. Keblis, "Release policies for assembly systems," *IIE Trans.*, vol. 27, pp. 507–518, 1995.

[10] I. Duenyas, "A simple release policy for networks of queues with controllable inputs," *Oper. Res.*, vol. 42, pp. 1162–1171, 1994.

[11] P. Glasserman and D. D. Yao, *Monotone Structure in Discrete-Event Systems*. New York: Wiley, 1994.

[12] C. R. Glassey, S. Seshadri, and J. G. Shanthikumar, "Production control of flowlines in the presence of multiple types of machine failures," Eng. Systems Research Center, Univ. California, Berkeley, rep. no. 93-15, 1993.

[13] ——, "Production control of flowlines: Characterization of optimal policies," Eng. Systems Research Center, Univ. California, Berkeley, rep. no. 93-16, 1993.

[14] C. R. Glassey and M. G. C. Resende, "Closed-loop job release control in VLSI circuit manufacturing," *IEEE Trans. Semiconduct. Manufact.*, vol. 1, pp. 36–46, 1988.

[15] P. R. Kumar, "Reentrant lines," *Queueing Syst.*, vol. 13, pp. 87–110, 1993.

[16] ——, "Scheduling semiconductor manufacturing plants," *IEEE Control Syst.*, pp. 33–40, Dec. 1994.

[17] R. C. Leachman, M. Solorzano, and C. R. Glassey, "A queue management policy for release of factory work orders," Eng. Systems Research Center, Univ. California, Berkeley, rep. no. 88-19, 1988.

[18] S. C. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, pp. 374–388, 1994.

[19] D. Mitra and J. B. Seery, "Dynamic adaptive windows for high speed data networks: Theory and simulations," in *Proc. ACM SIGCOMM*, Sept. 1990, pp. 30–37.

[20] R. Petrakian, "Shop floor control in wafer fabs using predictions and pricing," Eng. Systems Research Center, Univ. California, Berkeley, rep. no. 91-18, 1991.

[21] S. Seshadri and V. Srinivasan, "Procedures for estimating and guaranteeing delays in high speed communication networks," *The State of the Art in Performance Modeling and Simulation*, K. Bagchi, J. Walrand, and G. Zobrist, Eds. New York: Gordon and Breach, 1996.

[22] S. Seshadri, "Stochastic models of semiconductor fabs," Ph.D. dissertation, Walter A. Haas School of Business, Univ. California, Berkeley, 1993.

[23] S. Seshadri and F. Harche, "Bounds for the delay in some open queueing networks," unpublished.

[24] M. L. Spearman, D. L. Woodruff, and W. J. Hopp, "CONWIP: A pull alternative to Kanban," *Int. J. Prod. Res.*, vol. 28, pp. 879–894, 1990.

[25] G. Sullivan and K. Fordyce, "IBM Burlington's logistics management system," *Interfaces*, vol. 20, pp. 43–64, 1990.

[26] S. M. Sze, *VLSI Technology*, 2nd ed. New York: McGraw-Hill, 1988.

[27] R. Uzsoy, C. Lee, and L. A. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry, Part I: System characterization, performance evaluation and production planning," *IIE Trans.*, vol. 24, pp. 47–60, 1992.

[28] ——, "A review of production planning and scheduling models in the semiconductor industry, part II: Shop-floor control," *IIE Trans.*, vol. 26, pp. 44–55, 1994.
[29] M. H. Veatch and L. M. Wein, "Monotone control of queueing networks," *Queueing Syst.*, vol. 12, pp. 391–408, 1992.
[30] R. R. Weber and S. Stidham, "Optimal control of service rates in networks of queues," *Adv. Appl. Prob.*, vol. 19, pp. 202–218, 1987.
[31] L. M. Wein, "Scheduling semiconductor wafer fabrication," *IEEE Trans. Semiconduct. Manufact.*, vol. 1, pp. 115–120, 1988.
[32] W. E. Wilbert, "Research directions in electronics manufacturing," *IIE Trans.*, vol. 24, pp. 6–17, 1992.

C. Roger Glassey (M'88) received the B.S. degree in mechanical engineering from Cornell University, Ithaca, NY, in 1957, and completed the graduate program in industrial administration at the University of Manchester, U.K., as a Fulbright scholar. He received the M.S. degree in applied mathematics from the University of Rochester, Rochester, NY, in 1961, and the Ph.D. degree in operations research from Cornell University in 1965.

He served for three years as an officer in the U.S. Navy, then was with the Eastman Kodak Co., Rochester, NY, first as an Industrial Engineer, then in the Corporate Operations Research Group. He has been with the Department of Industrial Engineering and Operations Research, University of California, Berkeley, since 1965, and served as Chairman of the Department from 1980 to 1986. His primary research interests are production planning and scheduling, mathematical optimization, and microcomputers.

Jeyaveerasingam George Shanthikumar received the B.Sc. degree in mechanical engineering from the University of Sri Lanka, in 1972, and the M.Sc. and Ph.D. degrees in industrial engineering from the University of Toronto, Ont., Canada, in 1977 and 1979, respectively.

He is a Professor of Management Science, Haas School of Business, and a Professor of Industrial Engineering, Department of Industrial Engineering and Operations Research, University of California, Berkeley.

Dr. Shanthikumar was awarded the E.O.E. Pereira Gold Medal for the outstanding student, graduating from the Faculty of Engineering, University of Sri Lanka, 1972. He served as Vice-Chairman and Chairman, ORSA/TIMS Applied Probability Group, from 1990 to 1991 and 1991 to 1992, and as a Member, Council of the ORSA/TIMS Applied Probability Group from 1992 to 1994. He was an Associate Editor for the *Annals of Operations Research* (special volume, 1989), *International Journal of Flexible Manufacturing Systems*, *Journal of Discrete Event Dynamic Systems*, *Operations Research*, *Operations Research Letters*, *Probability in the Engineering and Informational Sciences*, and *Queueing Systems: Theory and Applications*.

Sridhar Seshadri received the Bachelor of Technology degree in mechanical engineering from the Indian Institute of Technology, Madras, in 1978, the Post Graduate Diploma in management from the Indian Institute of Management, Ahmedabad, in 1980, and the Ph.D. degree in management science from the University of California, Berkeley, in 1993.

He was an engineer with Jyoti, Ltd., Baroda, India, from 1980 to 1982, and from 1982 to 1984 was with Jaiprakash Associates, Baghdad, Iraq. He was a member of the faculty in the Operations Management Area, Administrative Staff College of India, from 1985 to 1989. He has been with the Department of Statistics and Operations Research, and the Operations Management Area, Stern School of Business, New York University, since 1993. His research interests are in the area of stochastic modeling and optimization, with applications to manufacturing, distribution, telecommunications, database design, and finance.

Dr. Seshadri is an Associate Editor of *Naval Research Logistics*.