

# Hierarchical Modeling and Control for Re-entrant Semiconductor Fabrication Lines: A Mini-Fab Benchmark\*

Kostas S. Tsakalis<sup>†</sup>

José-Job Flores-Godoy<sup>‡</sup>

Armando A. Rodriguez<sup>§</sup>

Arizona State University  
Department of Electrical Engineering  
System Science and Engineering Research Center  
Tempe, AZ 85287-7606

June 15, 1997

## Abstract

This paper addresses the problem of controlling re-entrant semiconductor fabrication lines. To focus the presentation, a virtual five-machine six-step mini-fab is used. This mini-fab – developed by Intel in collaboration with ASU – is intended to contain all of the features which make a real line difficult to control. Industry-like policies are discussed and compared with control-theoretic hierarchical policies.

## 1 Introduction

The control of the modern fab is one of the most challenging problems which has faced the semiconductor manufacturing industry. To remain competitive, it is essential for fab managers to implement *robust scheduling policies*; i.e. policies which robustly allocate fab resources (e.g. machines, operators, transport systems, etc.) in the presence of uncertainty (e.g. unreliable resources, etc.). Performance is typically measured using metrics associated with *throughput volume* and *throughput time*.

The modern fab is very complex. It involves many machines, process steps, and is highly *re-entrant* [5]; i.e. as wafers move from step-to-step, they often return to a given tool. As a consequence, the development of fab scheduling policies is nontrivial. It is

well known, for example, that scheduling problems are NP-complete [12, pp. ix].

Currently, fab policies are based on simple strategies (e.g. pull, LBFS, MBFS, push, etc.) obtained on the basis of experience, trial-and-error, simplified static models, and heuristics. Because fabs are so complex, operate in the presence of great uncertainty, and because the demand on fab resources has steadily increased, such policies often do not allow managers to reliably predict factory performance and fall short of optimizing performance. Moreover, they are often not robust to expected sources of uncertainty. Efforts to address such issues – as well as re-entrant issues – are still largely academic in that proposed methodologies are not yet readily applicable to real fabs [12]. This motivates the ideas put forward in this paper.

In this paper we describe a five-machine six-step re-entrant mini-fab whose purpose is to capture those essential features which make fab control difficult (e.g. batching, set-ups, etc.). The work presented in this paper is a consequence of ongoing interactions between semiconductor manufacturing practitioners from Intel Corporation and ASU faculty [2], [3], [6]-[8], [10], [11].

The remainder of this paper is organized as follows. Section 2 describes a five-machine six-step mini-fab developed by Intel [6] in collaboration with ASU. Hierarchical modeling issues are addressed in Section 3. In Section 4, policies which are similar to those found in real fabs are described. A simple pull policy and the effects of set-ups is considered in Section 5. This section further motivates the need for “intelligent” control-theoretic policies. Various control-theoretic policies are discussed in Section 6. Section 7 revisits some modeling issues. Finally, Section 8 summarizes the paper and presents directions for future research.

---

\*This research has been supported by the Intel Research Council; Sponsor: Dr. Karl Kempf. For more information see [www.intel.com/intel/other/research/](http://www.intel.com/intel/other/research/).

<sup>†</sup>Associate Professor. For more information see <http://enuxsa.eas.asu.edu/~tsakalis/> or contact [tsakalis@asu.edu](mailto:tsakalis@asu.edu).

<sup>‡</sup>Graduate Student. For more information contact [jjobfg@IMAP1.ASU.EDU](mailto:jjobfg@IMAP1.ASU.EDU).

<sup>§</sup>Associate Professor. For more information see <http://www.eas.asu.edu/~aar/> or contact [aar@asu.edu](mailto:aar@asu.edu).

## 2 Description of Mini-Fab

In this section, we describe the five-machine six-step re-entrant semiconductor fabrication mini-fab shown in Figure 1. This mini-fab was selected because it exhibits all of the essential features which make fab scheduling difficult. Such features include: re-entry, disparate processing times, batching, set-ups, preventive maintenance (PM), and emergency maintenance (EM). These features will all be considered throughout this paper – when EM is included it will be explicitly stated. Other features – not addressed in this paper – include: two products, test wafers, operators, technicians, and an unreliable transportation system. A detailed description of this mini-fab may be found in [6] or at <http://www.eas.asu.edu/~aar/>.

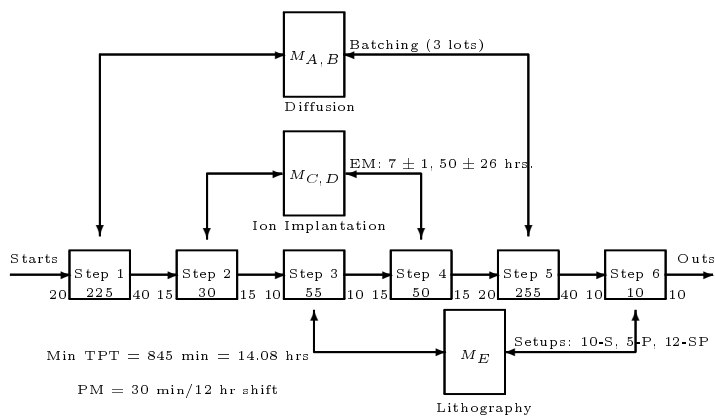


Figure 1: Canonical Step-Based Visualization of Five-Machine Six-Step Re-entrant Mini-Fab

Figure 1 shows that the mini-fab involves six processing steps: Steps 1 through 6. These steps are served by three stations containing five machines: Machines A, B, C, D, and E. Loading times, processing times, and unloading times are indicated in Figure 1 in minutes. The minimum time for an entering lot (start lot) to exit the line is 845 minutes (14.08 hrs.). Machines A and B reside within the first station, are identical, batch 3 lots at a time, and serve steps 1 and 5. Machines C and D reside within the second station, are identical, batch 1 lot at a time, and serve steps 2 and 4. These machines are unreliable in that they require emergency maintenance (EM) every  $50 \pm 26$  hours for  $7 \pm 1$  hours. Machine E resides within the third station, batches 1 lot at a time, and serves steps 3 and 6. This machine requires set-ups: 10 minutes for a step change, 5 minutes for a product change, and 12 minutes for a step and product change. In this paper, only step changes are considered. All machines require preventive maintenance (PM) – lasting 30 minutes – every 12 hour shift.

For this mini-fab, the bottleneck is Machine E and the bottleneck rate is given by:

$$\frac{1}{10 + 55 + 10 + 10 + 10 + 10} (720 - 30 - 20)$$

or 6.38 lots per shift. This rate assumes 30 minutes for PM and two 10 minute set-ups per 12 hour shift.

## 3 Hierarchical Modeling

This section discusses hierarchical modeling issues. To initiate the discussion, some perspective is provided.

Decisions within the modern fab are made hierarchically (see Figure 2). Hierarchical decision making is common whenever there exists a natural *time-scale separation* [4], [9].

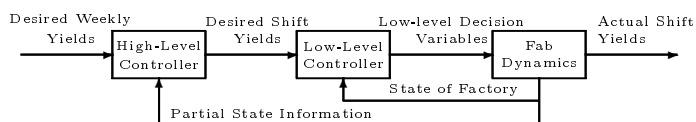


Figure 2: Visualization of Hierarchical Decision Making within the Modern Fab

Within a fab, the *frequency of events* is “low” at the highest level – where long planning horizons are involved – and “high” at the lowest level – where frequent decisions are made. At the highest level, desired per-week yields are determined on the basis of economic factors. These *target yields* are converted to desired per-shift yields by a high-level controller. A low-level controller then ensures that the appropriate low-level decisions are made so that the desired per-shift yields are achieved to the extent possible. It is the low-level controller which ultimately decides how resources are allocated within the factory. The above controllers are designed on the basis of high- and low-level models, respectively. We now briefly address high- and low-level modeling issues.

### 3.1 High-Level Modeling

At high-levels, decisions are made over long planning horizons. For such decision making, *flow models* involving “average variables,” such as those found in the literature are appropriate [1], [3], [8], [9]. Such models take on the following structure:

$$x_{n+1} = Ax_n + Bu_n \quad (1)$$

$$Cx_n + Du_n \leq 1 \quad (2)$$

$$f(x_n, u_n) \geq 0 \quad (3)$$

where the components of  $x_n$  denote average buffer levels and the components of  $u_n$  denote average start

rates and step utilizations. Such models have been justified primarily on the basis of empirical evidence [1] (see also Section 7). In [3], more quantitative reasons are given.

### 3.2 Low-Level Modeling

At low-levels, more accurate models may be required. Such models may be obtained by sampling the original discrete event system [3]. Doing so yields a model with a structure identical to that as the model above – with variables taking on discrete integer values. Such a model may be useful for simulation [2] and analysis. Future work will investigate its utility for design. Specifically, such models may be used as a starting point for the derivation of useful flow models. See <http://enuxsa.eas.asu.edu/~tsakalis/>.

## 4 Industrial-like Policies

Ongoing collaborations with Intel fab researchers have resulted in various industry-like controllers or policies [10]. These include 2 start release policies, 2 outer loop policies, and 6 inner loop (“pull until the shift goal is met”) policies. In this section, any combination of release, outer loop, and inner loop policy – and the associated 6 design parameters [10] – will be referred to as a “controller”. It should be noted that these controllers are “designed around the constraint” – in this case Machine E. In a real fab, the bottleneck may change. This provides one reason why “intelligent” control-theoretic policies are needed.

A small set of controllers (2500) were examined in the presence of PM and set-ups but in the absence of EM. The initial goal was to acquire performance limitation information with respect to product *output per shift* and *through-put-time* (TPT). In every case, the mini-fab simulator [2] was run for 300 shifts (150 days)<sup>1</sup> with the buffers – one for each step – initially empty. For each case, we determined the average lots produced per shift and the normalized average through-put-time per lot. A plot of one versus the other – containing these figures for all 2500 cases – is provided in Figure 3<sup>2</sup>. As expected, the figure does exhibit the presence of a concave *Pareto-boundary of performance* between the two measures. This boundary establishes a precise trade-off between fab per shift (maximum) output and (minimum) TPT. In this figure, a normalized average TPT of  $T$  corresponds to  $845(T + 1)$  minutes.

Next, we selected a subset (10) of the original (2500) controllers. These controllers were examined for (44)

<sup>1</sup>All simulations in this paper will take place over 300 shifts.

<sup>2</sup>Each point took 5 minutes to generate for a total CPU time of approximately 208 hours on a 90 Mhz Pentium PC.

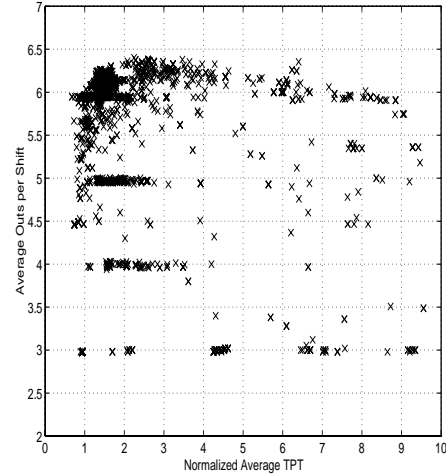


Figure 3: Industrial-like policies: 2500 different “controllers,” buffers initially empty.

different initial buffer conditions. The resulting plot – shown in Figure 4 – reveals that the sensitivity to initial conditions (ICs) could be very bad depending on the controller used. In what follows, we will find control-theoretic policies which exhibit less sensitivity to initial conditions.

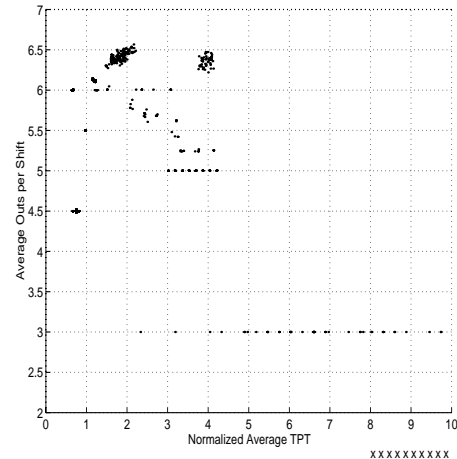


Figure 4: Industrial-like policies: 10 different “controllers,” 44 different ICs.

## 5 Instability in the Presence of Set-ups: A Simple Pull Policy

In this section, we further motivate the need for “intelligent” control-theoretic policies. Specifically, we show how a simple – commonly found – pull policy may result in instability because it does not handle set-ups well.

Specifically, we examine a simple pull policy for two different start release rates – both below the critical bottleneck rate of 6.38 lots per shift. The pull policy provides very good performance when set-ups are not included or when the release rate is well below the critical rate. If a 6.18 lot per shift release rate is used, one obtains the results shown in Figure 5. The figure shows that set-ups reduce system capacity severely. This manifests itself in the form of oscillations in buffer 3’s level.

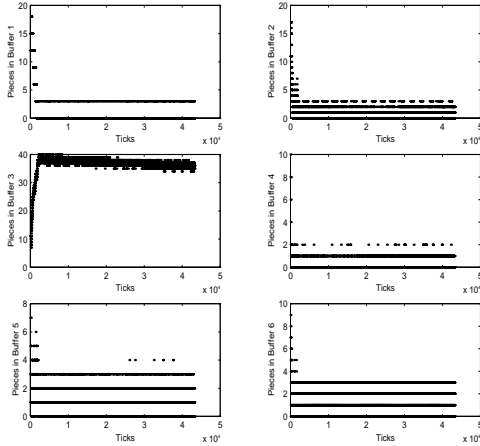


Figure 5: Buffer sizes for simple pull policy; start release rate: 6.18 lots per shift, Tick = 5 minutes.

If a 6.36 lot per shift release rate is used, one obtains the results shown in Figure 6. This figure shows that the system becomes unstable – see growth in buffer 3 level – because set-ups decrease system capacity below the 6.36 lots per shift release rate. This shows that simple policies may result in very bad performance. This further motivates the need for “intelligent” control-theoretic policies.

## 6 Control-Theoretic Policies

In this section control-theoretic policies are examined. Constant start release policies are used throughout – in contrast to the industrial-like policies examined earlier. An outer loop policy designed to minimize a Lyapunov function using one-step look-ahead information is used. The goal of the resulting constrained-minimization is to empty the buffers. The overall effect of the minimization is to produce as many outs as starts released into the fab and to empty any extra lots in the buffers. Two inner loop policies are examined: a fixed priority policy and a variable priority policy (VPP). Each policy is now considered.

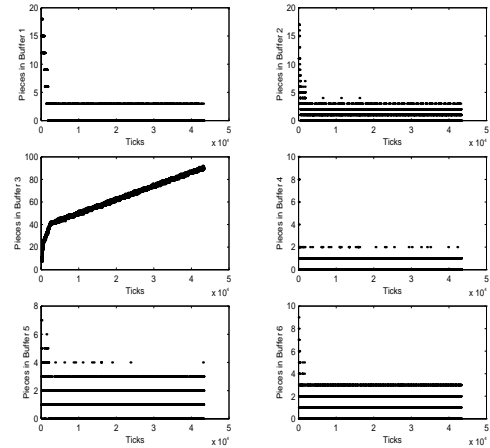


Figure 6: Buffer sizes for simple pull policy; start release rate: 6.36 lots per shift, Tick = 5 minutes.

### 6.1 Fixed Priority (Pull-Like) Inner Loop Policy

Using a fixed priority (pull-like) inner loop policy (FPP), we examined fab performance for different (constant) release rates and (44) different initial conditions. Figure 7 shows the results. This policy maintains steady-state performance close to the boundary but exhibits undesirable sensitivity to initial conditions for some release rates.

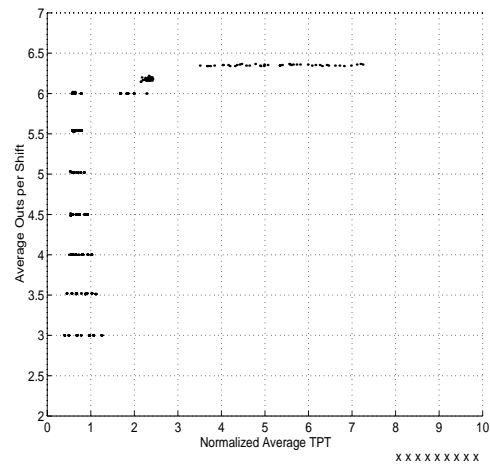


Figure 7: Constant start release rates, Lyapunov based one-step look-ahead minimization for outer loop, FPP inner loop, 44 ICs.

When the start release rates are small (3–4 lots/shift), the sensitivity with respect to initial conditions is larger than when the release rate is between 4.5–5.5 lots/shift. This may be attributed to batching and empty buffer saturation effects. When a release

rate of 6 lots/shift is used, the sensitivity with respect to initial conditions is much worse. This may be attributed to utilization saturation effects and bad set-ups patterns. Depending on the initial condition, the fab converges to a state in which the actual utilization for the bottleneck machines is decreased by the set-up time in such a way that only the goal of producing as many outs as starts released is fulfilled. For the case of 6.18 lots/shift this bad spread (sensitivity) is not present because the release rate does not permit a small periodic pattern. For the case of 6.36 lots/shift a large spread is seen, but it must be noted that the fab has not reached steady state – even after 300 shifts. Despite this, the buffers remain bounded and decreasing – in contrast to the simple pull policy considered in Section 5.

Next, we included emergency maintenance (EM). In this case the stochastic nature of the process removes the possibility of small periods and the sensitivity due to initial conditions improves as seen in Figure 8.

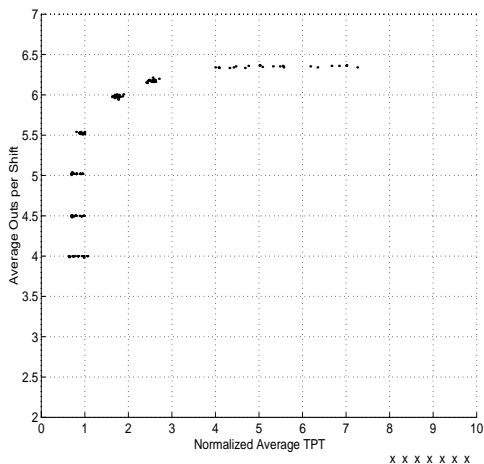


Figure 8: Constant start release rates, Lyapunov based one-step look-ahead minimization for outer loop, FPP inner loop, 22 ICs, and EM.

## 6.2 Variable Priority Inner Loop

Under the variable priority policy (VPP), the priority is allocated according to which step will require more utilization in order to achieve the shift goal. The priority is changed every time the outer loop is calculated (i.e. once per shift). With this inner loop policy, the effects of the utilization saturation and set-ups are lessened. Specifically, the sensitivity due to initial conditions is improved (see Figure 9) with respect to the previous (deterministic) case (see Figure 7); the best case got a little worse but the worst case got a lot better. The effect of empty buffers and batching is still evident for small release rates (e.g. 3–4 lots/shift). This policy maintains steady-state performance close to the

boundary without excessive sensitivity to initial conditions. However, the outputs show some – perhaps necessary – irregularity in their distributions.

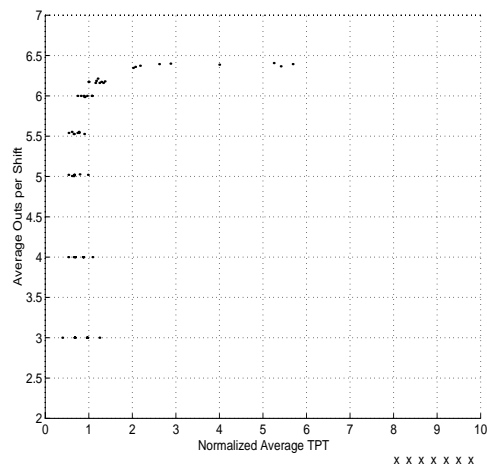


Figure 9: Constant start release rates, Lyapunov based one-step look-ahead minimization for outer loop, VPP inner loop, 9 ICs.

## 7 Modeling Issues Revisited: The Flow Model

In this section, we provide additional empirical evidence demonstrating the utility of flow models in approximating system behavior. To do this we used a constant start release policy. A Lyapunov based one-step look-ahead minimization was used for the outer loop. A variable priority policy (VPP) was used for the inner loop – see Section 6.2. The resulting overall closed loop behavior [2] was compared to that produced by an average flow model with a Lyapunov based feedback control law. The results for a 6.18 lots per shift release rate are seen in Figure 10.

During the transient the average closed loop flow model approximates the closed loop simulator. Once the steady state is reached the approximation is not as good due to a loss in resolution caused by batching. Even for a release rate of 6.36 lots per shift (see Figure 11), the approximation is good for those buffers whose levels are above the resolution due to batching and integer number of lots.

## 8 Summary

In this paper, a five-machine six-step mini-fab was described. This mini-fab contains all of the essential features which makes fab scheduling difficult. Hierarchical modeling and control issues were discussed.

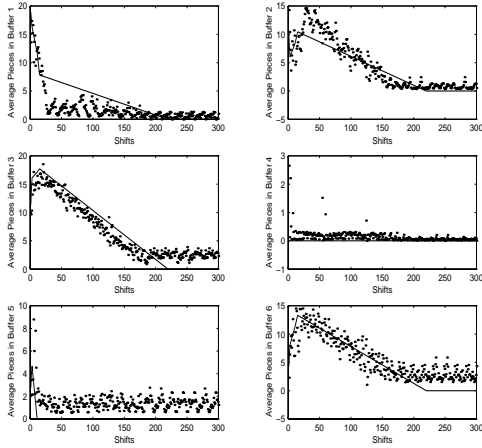


Figure 10: Average buffer sizes for closed loop flow model (solid) and closed loop “simulator” (dots). Start release rate: 6.18 lots per shift.

Industrial-like and control-theoretic scheduling policies were presented and compared. In conclusion, the results thus far indicate that control-theoretic policies may (i) be systematically obtained, (ii) result in a more predictable fab, and (iii) outperform commonly used industrial heuristics. Future work will address transient performance and other features described in [6].

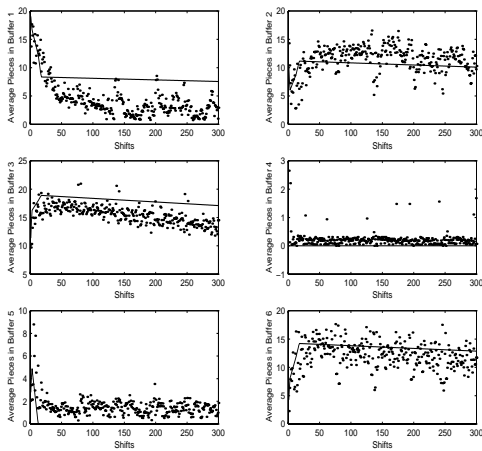


Figure 11: Average buffer sizes for closed loop flow model (solid) and closed loop “simulator” (dots). Start release rate: 6.36 lots per shift.

### Acknowledgments

The authors would like to thank Drs. K. Kempf and N. Srivatsan of Intel Corporation, for their support throughout the course of this work.

### References

- [1] R. Alvarez-Vargas, Y. Dallery, and R. David, “A Study of the Continuous Flow Model of Production Lines with Unreliable Machines and Finite Buffers,” *Journal of Manufacturing Systems*, Vol. 13, No. 3, 1994, pp. 221-234.
- [2] M. ElAdl and A.A. Rodriguez, “Five-Machine Six-Step Re-entrant Semiconductor Mini-Fab Simulator,” report, Arizona State University, 1995.
- [3] M. ElAdl, A.A. Rodriguez, and K. Tsakalis, “Hierarchical Modelling and Control for Re-entrant Semiconductor Fabrication Lines,” *Proceedings of the 36<sup>th</sup> IEEE Conference on Decision and Control*, Kobe, Japan, December,
- [4] S. Gershwin, “Hierarchical Flow Control: A framework for Scheduling and Planning Discrete Events in Manufacturing Systems,” *Proc. IEEE*, Vol. 77 1989, pp. 195-208.
- [5] P.R. Kumar, “Re-Entrant Lines,” Technical report, Coordinated Systems Laboratory, University of Illinois, Urbana, IL, received 1994.
- [6] K. Kempf, “Detailed Description of a Two-product Five-machine Six-step Re-entrant Semiconductor Manufacturing System,” prepared report, Intel Corporation, Technology & Manufacturing Group, August, 1994.
- [7] A.A. Rodriguez, “A Release Policy Governor (RPG) for Re-entrant Semiconductor Fabrication Lines,” submitted for publication in the *International Journal of Control*, 1997.
- [8] A.A. Rodriguez, K.S. Tsakalis, et. al., “Hierarchical Modeling and Control of Re-entrant Semiconductor Fabrication Lines,” Invited Session, *1997 American Control Conference*, Albuquerque, NM, 1997.
- [9] S.P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, Birkhäuser, 1994.
- [10] K.S. Tsakalis and J.J.F. Godoy, “Industry-like Policies for the Five-Machine Six-Step Mini-Fab Benchmark,” prepared notes, Department of Electrical Engineering, Arizona State University, August 1996.
- [11] B. Zaslavsky, A.A. Rodriguez, and D. Collins, “Time Optimal Control of Re-entrant Semiconductor Fabrication Lines,” submitted for publication in the *International Journal of Control*, 1997.
- [12] M. Zweben and M.S. Fox, *Intelligent Scheduling*, Morgan Kaufman Publishers, 1994.