

SIMULATION AS A PLANNING AND SCHEDULING TOOL FOR FLEXIBLE MANUFACTURING SYSTEMS

Glenn R. Drake
Jeffrey S. Smith
Brett A. Peters

Department of
Industrial Engineering
Texas A&M University
College Station, TX 77843, U.S.A.

ABSTRACT

This paper describes a flexible simulation technique which facilitates automated experimentation of different scheduling rules. An enhanced version of Arena/SIMAN is used to develop an extremely high fidelity model of the manufacturing system. The decision making of an FMS is characterized and a framework for simulation-based scheduling and control is also introduced.

1 INTRODUCTION

The emergence of flexible manufacturing systems (FMS) has sparked an increased interest and appreciation for real-time planning, scheduling, and control. An FMS is defined as a manufacturing system consisting of automatically reprogrammable machines (material processors), automated tool deliveries and changes, automated material handling and transport, and coordinated shop floor control (Askin and Standridge 1993). Real-time activities primarily refer to daily operations that require efficient, timely, and adaptive responses to short-term planning, scheduling, and execution problems. Pertinent areas of interest include job releases, loading sequences, dead-locks, and response to resource disruptions such as machine or tool failure.

Real-time control of an FMS is not a trivial task. Flexible routings, processing, and part mix, as well as the dynamic nature of a shop floor, place tremendous demands upon the control system. A detailed understanding of operational information, beyond mere "data crunching," is required for efficient production. Static schedules and strategies, developed ahead of time,

quickly lose validity in a rapidly changing system, and thus cannot be directly applied over long planning horizons. To make matters even more difficult, no two flexible manufacturing systems are identical, and system decision making can vary in both type and complexity. Due to the dynamic nature of FMS and inherent differences between systems, researchers argue that generic, optimal seeking solutions may be too difficult to resolve in real-time (Stecke and Solberg 1981; Wu 1987; Cho and Wysk 1993; Jones *et al.* 1995).

Recently, there has been increased interest in the use of *simulation* for real-time planning, scheduling, and control (Wu and Wysk 1989; Erickson *et al.* 1987; Harmonosky and Barrick 1988; Harmonosky 1990; Cho and Wysk 1993; Smith *et al.* 1994; Jones *et al.* 1995; Harmonosky and Robon 1995). Traditionally, simulation has been applied to long-term planning and design of manufacturing systems. These models have been termed "throw away models" because they are seldom used after the initial plans or designs are finalized (Thompson 1994). Recent reported applications of simulation for real-time, operational control include emulation of real-time control systems, adaptive scheduling and planning, real-time displays of system status, performance forecasting, as well as actual implementation into a shop floor controller (Smith *et al.* 1994; Jones *et al.* 1995). The use of simulation has appeared favorable to purely analytical methods which often fail to capture complex interactions of a particular FMS (Stecke and Solberg 1981; Erickson *et al.* 1987; Wu and Wysk 1989).

Applying simulation as a real-time tool requires insight into the responsibilities of the simulation model and its role within the shop floor control system. The

objective of this paper is to represent the problems and decision making requirements for an FMS so that real-time operations of a system may be effectively modeled. A framework for applying simulation models to on-line planning, scheduling, and control problems is presented. This framework uses the *multi-pass* scheduling approach (Wu and Wysk 1989) and a modified *Arena/SIMAN* simulation platform for implementation (Smith *et al.* 1994).

This paper is organized as follows: Section 2 examines previous research in the multi-pass and simulation control areas. Section 3 outlines a methodology for representing system problems and decision making. The concept of "flexible simulation" and an architecture for exploiting multi-pass analysis and real-time control using *Arena/SIMAN* software is described in Section 4. Section 5 offers conclusions as well as future research directions in this area.

2 PREVIOUS RESEARCH

The scheduling problems encountered in an FMS can be separated into several distinct types which encompass a wide range of resources including parts, robots, machines, and AGVs. Stecke and Solberg (1981) categorize different scheduling problems and apply sets of dispatching rules to each problem in an effort to evaluate the impact of various rules on the system performance. Several researchers have since evaluated different problems under different sets of rules. Egbelu and Tanchoco (1984) examine scheduling problems and dispatching rules for AGVs. Tang *et al.* (1995) examine one particular system and define six scheduling problems associated with this system. The impact of a "good" schedule for a particular problem and the effect of any one dispatching rule has been found to vary with several factors such as system layout, system state, and the desired performance measure (Dar-El and Wysk 1982; Wu and Wysk 1989; Cho and Wysk 1993; Tang *et al.* 1995; Jones *et al.* 1995).

Studies have shown that combinations of dispatching rules over a system's production cycle can produce better performance than a single rule alone (Wu and Wysk 1989; Maturra *et al.* 1993; Cho and Wysk 1993). Rapidly changing environments and dynamic flows of jobs can erode the advantages of a particular plan or strategy over time. For example, Yamamoto and Nof (1985) showed that a

scheduling/rescheduling approach improved system performance 2-7% compared with fixed dispatching procedures when just machine breakdowns were taken into account.

A Multi-Pass Expert Control System (MPECS) is an approach that uses multi-pass scheduling to dynamically resolve factory scheduling problems (Wu 1987). MPECS breaks up a production cycle into several periods separated by decision points. At each decision point, the current factory status is evaluated and a series of deterministic simulations are run to see which control rule from a set of alternative rules (generated by the expert system) provides the best performance. The rule is then implemented by the control system. Wu found that MPECS consistently outperformed a single rule for part scheduling and that the length of the simulation window is an important factor for analysis.

Other researchers (Cho and Wysk 1993; Jones *et al.* 1995) have suggested using neural networks to identify candidate rules for multi-pass simulation analysis. Cho (1993) defines five types of scheduling problems in the context of an automated workstation. At each decision point, the neural network generates candidate rules for each problem type and these rules are then evaluated through simulation. Jones *et al.* (1995) take into account multi-criteria performance measures. When a new schedule is desired, a neural net generates good rules for each performance measure and then simulation is used to predict how each rule does against all performance measures simultaneously. In both cases, the neural network is trained off-line by the simulation under a variety of input conditions.

Smith *et al.* (1994) examine the application of discrete event simulation for shop floor control of a flexible manufacturing system. In this pretense, simulation is used not only as an analysis and evaluation tool, but also as a task generator to control the physical equipment on the shop floor. Figure 1 illustrates the RapidCIM shop floor control architecture. The Task Generator (*Arena/SIMAN*) and the execution module communicate through a *task initiation queue* (TIQ) and a *task completion queue* (TCQ). This explicit separation of decision making and execution introduces flexibility into the SFCS and allows "plug and play" analysis. However, the paper concentrates on the technological aspects of the problem, and additional experimentation into the actual use of simulation as an on-line planning and scheduling tool is required.

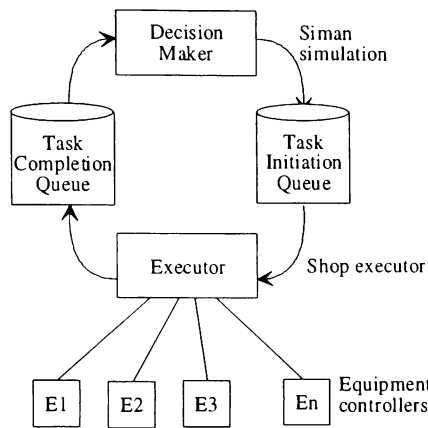


Figure 1: RapidCIM Simulation-Based Control

From previous research, three points have become apparent with regard to on-line simulation:

1. There are several scheduling problems and associated dispatching rules to consider in FMS.
2. Both the multi-pass approach to real-time scheduling and planning, and the use of simulation as a combination analysis and control tool appear promising.
3. Separation of planning, scheduling, and execution functions introduces flexibility to a SFCS, and this implies that on-line simulation models should also have a clear separation between decision making and the physical characteristics of the system.

Section 3 outlines a methodology for problem representation within an FMS so that a flexible, on-line simulation can be constructed.

3 REPRESENTATION OF FMS PROBLEMS AND DECISION MAKING

Complex, real-time decision making is a common aspect of all flexible manufacturing systems, irrespective of the layout or configuration. In general, FMS planning, scheduling, and execution problems are directly attributable to factors such as layout, number of resources, complexities of the transporter network, variety of the product mix, managerial objectives, etc. However, the problems themselves usually vary from system to system. A *problem set* is a set of real-time decisions encountered during operation. Issues might include part scheduling on machines, AGV dispatching,

or task scheduling of material handling equipment. For this paper, a problem set is comprised of system decision making that could benefit from real-time simulation analysis.

A simple methodology is presented next for defining a problem set and decision making within any FMS. An example FMS is presented for discussion. A broad range of problem types can be applied under the proposed methodology. Problem definition and decision representation is a key ingredient for on-line “plug and play” analysis.

3.1 Example FMS

An example FMS is shown in Figure 2. Material processing is performed on 4 CNC machines which are serviced by three robots. The material transport system consists of a conveyor system and specialized pallets for part routing. Inventory is stored at an I/O station and part buffers at each workstation. This system is similar to a flexible manufacturing lab in current development at Texas A&M University.

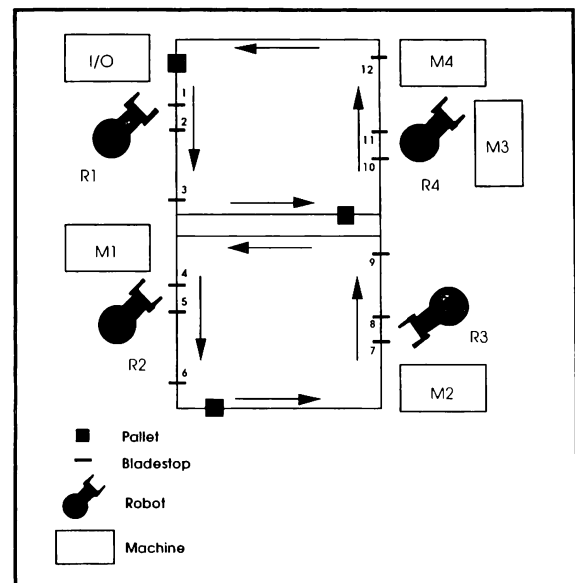


Figure 2: Example FMS

3.2 Defining the Problem Set for the Example

An important step in simulation modeling for on-line application involves identifying the set of real-time (operational) problems that could benefit from simulation analysis and locating the decision points of those problems within the system.

The following problem set might be considered reasonable for the example FMS in Figure 2:

1. Part Dispatching Problem: Given a set of contending parts in a queue, which part leaves the queue next?
2. Transporter Request Problem: Given multiple requests for pallets in the system, which part request is serviced next?
3. Material Handling Problem: Given a set of parts vying for an available material handler (i.e., robot), which part receives the handler next?
4. Intersection Contention Problem: Given a set of pallets waiting for access to an intersection (i.e., lift and transfer), which pallet accesses the intersection next?
5. Alternative Selection Problem: Given a set of feasible alternatives (i.e., next machine, process, etc.), which alternative does a part select next?

The above questions represent only a portion of scheduling issues that might be considered when constructing a problem set. Some of the above mentioned problems may be trivial with respect to other systems due to layout, configuration, etc. There may be problems associated with a system that are not defined for simulation analysis due to management policy or lack of significant effect. For example, the number of pallets in the system or lot sizes of the orders may be fixed and thus eliminated from real-time decision making. Also, off-line analysis with a simulation model might have shown a particular scheduling problem robust such that the cost of on-line adaptation or "play" is not merited. In the example FMS shown, it is likely a small number of pallets would render problem 4 insignificant, allowing the use of fixed or hard-coded rules in the simulation model.

The "decision points" in a particular system occur at locations or *incidences* where scheduling problems are encountered. Associating different incidences to different scheduling problems has been performed by previous researchers including Cho (1993). At an incidence, several *rules* may be applicable for obtaining problem solutions. A vector notation can be used to define the decision structure for scheduling problems within a system. In this notation, each vector component corresponds to an incidence and the value of that component represents the incidence's scheduling rule.

Definition: A *decision vector* is a vector $\bar{a} = (a_1, a_2, \dots, a_n)$, where n is the number of incidences and a_i can take up to k_i values.

The components of a *decision vector* might represent specific equipment, physical areas in the system, such as intersections or queues, or system states such as multiple part requests for a resource. Applicable scheduling rules for incidences could be analytical, heuristic, derived from state tables, user-customized etc. In general, a decision vector encompasses the overall strategy for obtaining solutions to real-time problems and could be defined for any flexible manufacturing system. This simple concept completes the first step towards modeling simulations intended for on-line use.

As an example, the next section further describes the first problem of the problem set defined above. Example incidences, rules, decision vectors, and possible Arena/SIMAN modeling constructs are discussed.

3.2.1 Part Dispatching Problem

The part dispatching problem concerns the sequencing of parts from a resource queue to either a machine, material handler, or the system as a whole. In the example FMS, there are five queues or "incidences" where this problem could occur: the queues at the I/O station (system entry) and each of the machines.

Decision Vector: The vector components in this case are the five system queues (I/O Queue, machine queues 1 through 4) and the values of those components represent the applied rules. For example, one possible vector of scheduling strategies could be shown as:

$$\bar{a} = (1, 3, 2, 2, 1),$$

where the corresponding value for each scheduling rule is defined as follows:

$a_i =$	1	2	3
Rule	SPT	FIFO	EDD

Thus, for the decision vector defined above, queue number 1 (the I/O queue) uses rule number 1 (SPT) to select the next part, queue number 2 (M1 queue) uses EDD etc. Although three relatively simple dispatching rules have been shown, rules which take into account other global or local conditions (such as setup times) could also be included as possibilities for relevant incidences in the system.

SIMAN constructs for modeling using decision vectors: A real-time simulation model should incorporate constructs such that different decision vectors (i.e., dispatching rules) for the part dispatching problem can be applied across the system without model recompilation. This provides the necessary flexibility for on-line analysis and control. Experimentation is also simplified for those unfamiliar with the simulation language. The model fragment below shows how this can be done for the example FMS using SIMAN. The ranking rule for each queue in the system is defined as LVF (priority) in the experimental frame. Before entering a queue in the model frame, an entity is assigned a priority based on the value of a global variable. For example, before entering the entry queue at the I/O station, the entity is assigned priority according to global variable *IOQ*. This variable effectively dictates the expression in the experimental frame (i.e., dispatching rule) and entity queue rank. Thus, *IOQ*=‘1’ means SPT, ‘2’ means FIFO, etc.

```

Model Fragment:
Model Frame:
ASSIGN: Priority = IOQRules(IOQ);
:Assigns priority based on IOQ
QUEUE, IOQueue;           The queue for the I/O station
Experimental Frame:
QUEUE, 1, IOQueue, LVF(Priority);
EXPRESSIONS: IOQRules(3),
Next_ProcessingTime,      !Shortest Processing Time
TNOW,                     !First In, First Out
DueDate;                  !Earliest Due Date
VARIABLES: IOQ;          The Global Variable
    
```

3.2.2 Experimentation Using Sets of Decision Vectors

In the SIMAN model fragment above, a global variable (*IOQ*) dictated rule selection for a particular decision point in the system. Global SIMAN variables could be similarly defined for other scheduling problems and their incidences. A vector of these global variables is called a decision vector. Vector components thus represent real-time decision points and the range of values for each component define applicable rules.

By using decision vectors to define the decision making of a system, scheduling scenarios are easily constructed. The simulation model can be used to “plug and play” with different scheduling strategies *without* recompilation or remodeling. This provides a convenient and flexible method for both off-line and on-line testing of rules which could affect system performance.

A set of decision vectors can be used to define a scheduling experiment. *Batches* of strategies are thus

compared via simulation analysis in an automated fashion. An experiment for the example FMS is shown below. The simulation model READs and simulates each vector of scheduling strategies line by line from an input data file. The results of each scenario, such as part flowtimes or number of late jobs, are returned accordingly.

```

/*Scheduling Experiment*/
1 2 2 1 3 /*Vector for Scheduling Strategy #1*/
2 2 2 2 1 /*Vector for Scheduling Strategy #2*/
3 1 1 2 3 /*Vector for Scheduling Strategy #3*/
    
```

4 IMPLEMENTING ON-LINE SIMULATION

4.1 A Flexible Simulation

A *flexible simulation* model separates the physical characteristics of a system from real-time scheduling, planning, and control problems so that different combinations of decision strategies can be readily “plugged in” and evaluated without model recompilation. This type of adaptivity is necessary for efficient use in dynamic, on-line environments where the user is often another piece of software or someone unfamiliar with the simulation language. Unarguably, real-time use of a simulation model presents different challenges than those encountered in traditional off-line design and analysis. As a result, new modeling approaches and further language development may be necessary to effectively exploit this concept.

Section 3 defined a methodology for representing real-time decision making and problems encountered in an FMS. The goal was to provide a representation such that a given system’s decision structure could be easily separated from the description of the physical system. This provides the ability to overlay scheduling scenarios onto a system for either analysis or control. It was shown how, through vectors of global variables, a system’s decision structure could be defined. Multi-pass scheduling and planning thus concerns the evaluation of alternative strategies or vectors.

Scheduling rules, developed separately and linked to a simulation model, can gain access to valuable dynamic knowledge such as current system state, times, etc. if the simulation is controlling the facility or operating in parallel. The combination of simulation and analytical or heuristic tools could be a powerful analysis aid for the future. The flexible simulation concept provides one step for constructing simulation models intended for real-time use.

4.2 Flexible Simulation Architecture Using Arena/SIMAN

Figure 3 shows the architecture by which a real-time simulation system can be established for on-line analysis and control using Arena/SIMAN software. Notice that the figure contains two identical simulation models. One is used primarily as a task generator to the shop floor (real-time clock and operation) while the other is used for analysis purposes. Although only one analysis model is shown, there could feasibly be several. The execution between the real-time model and the shop floor is identical to that developed by the RapidCIM project (Figure 1). Two different simulation models for analysis and control allow simultaneous task dispatching and schedule comparison. Otherwise, the shop floor might be forced to “shut down” during an analysis phase in the production cycle.

The “Planner/Scheduler” represents an entity above the simulation models. This function (an application, user, etc.) declares the scheduling experiments (file *Map.dat*), releases production orders (file *Orfile.dat*), defines run parameters (window length, time considered for output analysis, etc.), executes commands to the simulations through their run controllers, and gathers simulation output (file *Out.dat*). It is the planner/scheduler which dictates objectives and appropriate schedules to dispatch or analyze. Output analysis tools would appropriately be found within this module but are not shown here.

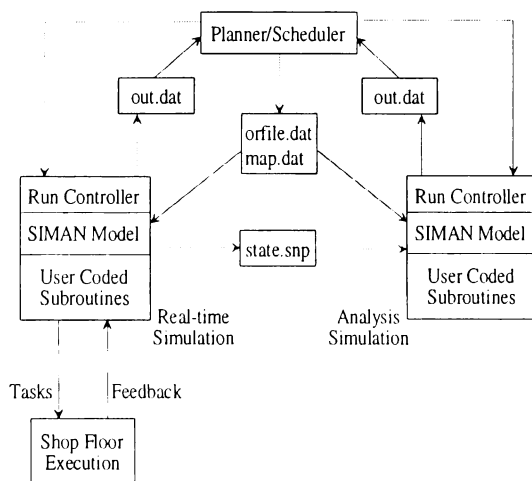


Figure 3: Proposed Simulation Architecture

4.2.1 File Types

As stated above and shown in Figure 3, there are four major file types for data transfer between the simulation

and scheduler/planner entities. Each are briefly described below:

1. *Map.dat* is a file or database which defines scheduling experiments for analysis.
2. *Orfile.dat* is a file or database for transfer of production orders into the simulation models. This file might include information such as the order number, order quantity, process plan, due date, etc.
3. *Out.dat* is an output file from which the scheduler gathers both real-time feedback from the control simulation and results of simulation analysis.
4. *State.snp* is a file which contains the current system state of a simulation. This file can be SAVED and RESTORED through run controller commands (see next paragraph).

4.2.2 Interactive Run Control

Arena/SIMAN maintains a command driven run controller which allows monitoring and interactive control of a simulation's execution. The scheduler/planner can use a variety of run controller commands as well as an interrupt to manipulate the on-line simulations. These commands can be found in Pegden *et al.* (1990) and a few are described below:

1. SAVE FileSpec. This command saves a snapshot of the system to the file *FileSpec* at some point in simulated time. It can be recalled with the RESTORE command.
2. EVENT Number. This command executes the user-coded sub-routine *Number*.
3. GO or GO UNTIL Time. Begins or resumes simulation execution till either *Time* or another interrupt occurs.

Suppose sub-routine 101 reads a new schedule from *Map.dat* and sub-routine 102 outputs performance data to *Out.dat*. If, at a discrete decision point ($Time = 300$), the planner/scheduler decided to evaluate a different schedule forward 120 minutes to see if higher performance could be achieved, the following might be the sequence of commands issued to the simulation run controllers (Sim #1 = Real-Time Model, Sim #2 = Analysis Model):

INTERRUPT (to Sim #1)	/Halts Simulation 1/
SAVE State.snp (to Sim #1)	/Saves Current State/

RESTORE State.snp (to Sim #2)	/Restores the Current State to Sim #2/
EVENT 101 (to Sim #2)	/Reads new scenario from Map.dat /
GO UNTIL 420 (to Sim #2)	/Simulates forward 120 minutes/
EVENT 102 (to Sim #2)	/Outputs performance to Out.dat /
EVENT 101 (to Sim #1)	/Scheduler likes the new option/
GO (Sim #1)	/Execute new decision structure/

The potential of additional run controller commands to enhance interactive execution of a simulation model is currently being investigated. Standard macros of run controller commands might be useful for facilitating scheduling and control applications.

5 CONCLUSIONS

Difficult planning, scheduling, and control problems for flexible manufacturing systems have sparked recent interest in simulation as an on-line tool. Traditionally, simulation has been applied to long-term planning or design. These models are "throw away models" in that they are seldom used after the associated plans or designs have been finalized. An on-line simulation model, however, is intended for daily use during manufacturing. The dynamic and random nature of a shop floor requires a higher flexibility and adaptiveness as compared to traditional, off-line use. Both new modeling approaches and further language development may be necessary to effectively exploit this field.

There is a need for representation of problems and decision making within flexible manufacturing systems so that a real-time simulation model can be efficiently constructed. Though complex decision making is a common aspect of flexible manufacturing, the problems encountered often vary from system to system. Adequate problem definition and decision representation is a key ingredient for flexible simulation modeling and analysis.

Previous research has shown promise in both multi-pass scheduling as well as the use of simulation as a combination analysis and control tool. Computational times of the multi-pass method will have to be solved for it to become a viable, real-time technique. It has been shown that a separation of planning, scheduling, and execution functions introduces flexibility to a SFCS. This would imply that on-line simulation should also have a clear separation between decision making and the physical characteristics of the system.

REFERENCES

- Askin, R. G., and C. R. Standridge. 1993. *Modelling and Analysis of Manufacturing Systems*. John Wiley & Sons, Inc., New York.
- Cho, H. 1992. An Intelligent Workstation Controller for Computer Integrated Manufacturing. PhD Dissertation, Industrial Engineering, Texas A&M University, College Station, TX.
- Cho, H., and R. A. Wysk. 1993. A Robust Adaptive Scheduler for an Intelligent Workstation Controller. *International Journal of Production Research*, Vol. 31, No. 4, 771-789.
- Dar-El, E. M., and R. A. Wysk. 1982. Job Shop Scheduling - A Systematic Approach. *Journal of Manufacturing Systems*, Vol. 1, 77-88.
- Erickson, C., Vandenberg, A., and T. Miles. 1987. Simulation, Animation, and Shop-Floor Control. In *Proceedings of the 1987 Winter Simulation Conference*, ed. H. Grant, W. D. Kelton, A. Thesen, 649-652. Atlanta, GA.
- Harmonosky, C. 1990. Implementation Issues Using Simulation for Real-Time Scheduling, Control, and Monitoring. In *Proceedings of the 1990 Winter Simulation Conference*, ed. R. P. Sadowski, R. E. Nance, O. Balci, 595-598. New Orleans, LA.
- Harmonosky, C. M., and D. C. Barrick. 1988. Simulation in a CIM Environment: Structure for Analysis and Real-Time Control. In *Proceedings of the 1988 Winter Simulation Conference*, ed. P. L. Haigh, J. C. Comfort, M. A. Abrams, 704-711. San Diego, CA.
- Harmonosky, C. M., and S. F. Robohn. 1991. Real-Time Scheduling in Computer Integrated Manufacturing: A Review of Recent Literature. *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 6, 331-340.
- Jones, A., Rabelo, L., and Y. Yuehwern. 1995. A Hybrid Approach for Real-Time Sequencing and Scheduling. *International Journal of Computer Integrated Manufacturing*, Vol. 8, No 2, 145-154.
- Law, A. M., and W. D. Kelton. 1991. *Simulation Modeling & Analysis*. McGraw-Hill Inc., New York.
- Matsuura, H., Tsubone, H., and M. Kanezashi. 1993. Sequencing, Dispatching, and Switching in a Dynamic Manufacturing Environment. *International Journal of Production Research*, Vol. 31, No.7, 1671-1688.
- Pegden, C. D., Shannon, R. E., and R. P. Sadowski. 1990. *Introduction to Simulation Using SIMAN*. McGraw-Hill, Inc., New York.
- Smith, J. S., Wysk, R. A., Sturrock, D. T., Ramaswamy, S. E., Smith, G. D., and S. B. Joshi. 1994. Discrete Event Simulation for Shop Floor Control. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, M. S. Manivannan, D. A. Sadowski, A. F. Seila, 962-969. Orlando, FL.

- Stecke, K. E., and J. J. Solberg. 1981. Loading and Control Procedures for Flexible Manufacturing Systems. *International Journal of Production Research*, No. 19, 481-490.
- Thompson, M. B. 1994. Expanding Simulation Beyond Planning and Design. *Industrial Engineering*, Vol. 26, No. 10, 65-67.
- Wu, D. S. 1987. An Expert System Approach for the Control and Scheduling of Manufacturing Cells. PhD Dissertation, Department of Industrial and Management Systems Engineering, Pennsylvania State University, University Park, PA.
- Wu, D. S., and R. A. Wysk. 1989. An Application of Discrete-Event Simulation to On-Line Control and Scheduling in Flexible Manufacturing. *International Journal of Production Research*, Vol. 27, 1603-1624.
- Yamamoto, M., and S. Nof. 1985. Scheduling and Rescheduling in the Manufacturing Operating System Environment. *International Journal of Production Research*, Vol. 23, 703-713.

AUTHOR BIOGRAPHIES

GLENN R. DRAKE is a research assistant in the Industrial Engineering Department at Texas A&M University. He received his B.S.I.E. in 1994 from Texas A&M University and is currently working on his M.S.I.E. His interests are in simulation-based planning, scheduling, and shop floor control.

JEFFREY S. SMITH is an assistant professor in the Industrial Engineering Department at Texas A&M University. His research interests are in shop floor control, manufacturing systems design, analysis and control, and simulation. He is an active member of IIE, SME, and IEEE.

BRETT A. PETERS is an assistant professor in the Industrial Engineering Department at Texas A&M University. His research interests are in manufacturing systems design, analysis and control, and simulation. He is an active member of IIE, SME, and INFORMS.