Manufacturing and Automation Systems: Techniques and Technologies

Part 3 of 5

CONTROL AND
DYNAMIC SYSTEMS

*Advances in Theory
and Applications*

Volume 47

CONTRIBUTORS TO THIS VOLUME

WARREN J. BOE
P. BUTRYN
LUIS C. CATTANI
CHUNHUNG CHENG
ALAN A. DESROCHERS
FRANK DICESARE
PAUL J. EAGLE
JOONG-IN KIM
VASSILIS S. KOUIKOGLOU
J. KRAMER
ANDREW KUSIAK
J. MAGEE
Y. NARAHARI
YANNIS A. PHILLIS
R. RAM
IN-KYU RO
M. SLOMAN
EDWARD SZCZERBICKI
N. VISWANADHAM
J. M. WILSON
BIN WU
S. DAVID WU
RICHARD A. WYSK

CONTROL AND DYNAMIC SYSTEMS

ADVANCES IN THEORY AND APPLICATIONS

Edited by

C. T. LEONDES

School of Engineering and Applied Science
University of California, Los Angeles
Los Angeles, California
and
College of Engineering
University of Washington
Seattle, Washington

VOLUME 47: MANUFACTURING AND AUTOMATION SYSTEMS: TECHNIQUES AND TECHNOLOGIES
Part 3 of 5

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich, Publishers
San Diego New York Boston
London Sydney Tokyo Toronto

ACADEMIC PRESS RAPID MANUSCRIPT REPRODUCTION

This book is printed on acid-free paper. ♾

Academic Press, Inc.
San Diego, California 92101

United Kingdom Edition published by
Academic Press Limited
24–28 Oval Road, London NW1 7DX

Library of Congress Catalog Number: 64-8027

International Standard Book Number: 0-12-012747-4

PRINTED IN THE UNITED STATES OF AMERICA
91 92 93 94 9 8 7 6 5 4 3 2 1

CONTENTS

CONTRIBUTORS vii
PREFACE ix

Techniques in Modeling and Control Policies
for Production Networks 1

Yannis A. Phillis and Vassilis S. Kouikoglou

Scheduling, Optimization and Control in Automated Systems 51

S. David Wu and Richard A. Wysk

Performability of Automated Manufacturing Systems 77

N. Viswanadham, Y. Narahari, and R. Ram

Modeling, Control, and Performance Analysis of Automated
Manufacturing Systems Using Petri Nets 121

Frank Dicesare and Alan A. Desrochers

Evalcon: A System for Evaluation of Concurrency
in Engineering Design 173

Andrew Kusiak and Edward Szczerbicki

Concurrent Engineering: Design of Assemblies for Schedulability 221

Andrew Kusiak

Multi-Criteria Optimization and Dynamic Control Methods in Flexible Manufacturing Systems 241

In-Kyu Ro and Joong-In Kim

Computer-Aided Manufacturing Systems Design: Framework and Tools 281

Bin Wu

Software Configuration Techniques in Operational Systems 325

M. Sloman, J. Kramer, J. Magee, and P. Butryn

Software for Dynamic Tool Modeling in Manufacturing and Automation 357

Luis C. Cattani and Paul J. Eagle

Techniques for Optimal Operation Allocation Methods in Manufacturing Systems—A Review of Non-Probabilistic Approaches 383

J. M. Wilson

A Branch-and-Bound Algorithm for Solving the Machine Allocation Problem 399

Chunhung Cheng, Andrew Kusiak, and Warren J. Boe

INDEX 421

CONTRIBUTORS

Numbers in parentheses indicate the pages on which the authors' contributions begin.

Warren J. Boe (399), *Intelligent Systems Laboratory, Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa 52242*

P. Butryn (325), *BP Research, BP International Ltd., Sunbury Research Centre, Sunbury on Thames, Middlesex TW16 7LN, England*

Luis C. Cattani (357), *Department of Mechanical Engineering, University of Detroit Mercy, Detroit, Michigan 48221*

Chunhung Cheng (399), *Intelligent Systems Laboratory, Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa 52242*

Alan A. Desrochers (121), *Electrical, Computer, and Systems Engineering Department, Center for Intelligent Robotic Systems for Space Exploration, Rensselaer Polytechnic Institute, Troy, New York 12180*

Frank Dicesare (121), *Electrical, Computer, and Systems Engineering Department, Center for Manufacturing Productivity and Technology Transfer, Rensselaer Polytechnic Institute, Troy, New York 12180*

Paul J. Eagle (357), *Department of Mechanical Engineering, University of Detroit Mercy, Detroit, Michigan 48221*

Joong-In Kim (241), *CIMS Laboratory, Department of Industrial Engineering, Hanyang University, Seoul 133-791, Korea*

Vassilis S. Kouikoglou (1), *Department of Production Engineering and Management, Technical University of Crete, Chania 731 00, Greece*

J. Kramer (325), *Department of Computing, Imperial College of Science Technology and Medicine, London SW7 2BZ, England*

Andrew Kusiak (173, 221, 399), *Intelligent Systems Laboratory, Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa 52242*

J. Magee (325), *Department of Computing, Imperial College of Science Technology and Medicine, London SW7 2BZ, England*

Y. Narahari (77), *Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India*

Yannis A. Phillis (1), *Department of Production Engineering and Management, Technical University of Crete, Chania 731 00, Greece*

R. Ram (77), *Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India*

In-Kyu Ro (241), *CIMS Laboratory, Department of Industrial Engineering, Hanyang University, Seoul 133-791, Korea*

M. Sloman (325), *Department of Computing, Imperial College of Science Technology and Medicine, London SW7 2BZ, England*

Edward Szczerbicki (173), *Intelligent Systems Laboratory, Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa 52242*

N. Viswanadham (77), *Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India*

J. M. Wilson (383), *Loughborough University of Technology, Loughborough, Leicestershire LE11 3TU, England*

Bin Wu (281), *Department of Manufacturing and Engineering Systems, Brunel University, Uxbridge, Middlesex UB8 3PH, England*

S. David Wu (51), *Department of Industrial Engineering, Lehigh University, Bethlehem, Pennsylvania 18015*

Richard A. Wysk (51), *Department of Industrial Engineering, Texas A&M University, College Station, Texas 77843*

PREFACE

At the start of this century, national economies on the international scene were, to a large extent, agriculturally based. This was, perhaps, the dominant reason for the protraction, on the international scene, of the Great Depression, which began with the Wall Street stock market crash of October 1929. In any event, after World War II the trend away from agriculturally based economies and toward industrially based economies continued and strengthened. Indeed, today, in the United States, approximately only 1% of the population is involved in the agriculture industry. Yet, this small segment largely provides for the agriculture requirements of the United States and, in fact, provides significant agriculture exports. This, of course, is made possible by the greatly improved techniques and technologies utilized in the agriculture industry.

The trend toward industrially based economies after World War II was, in turn, followed by a trend toward service-based economies; and, in fact, in the United States today roughly 70% of the employment is involved with service industries, and this percentage continues to increase. Nevertheless, of course, manufacturing retains its historic importance in the economy of the United States and in other economies, and in the United States the manufacturing industries account for the lion's share of exports and imports. Just as in the case of the agriculture industries, more is continually expected from a constantly shrinking percentage of the population. Also, just as in the case of the agriculture industries, this can only be possible through the utilization of constantly improving techniques and technologies in the manufacturing industries. As a result, this is a particularly appropriate time to treat the issue of manufacturing and automation systems in this international series. Thus, this is Part 3 of a five-part set of volumes devoted to the most timely theme of "Manufacturing and Automation Systems: Techniques and Technologies."

The first contribution to this volume is "Techniques in Modeling and Control Policies for Production Networks," by Yannis A. Phillis and Vassilis S. Kouikoglou. Large, interconnected systems are widely used in manufacturing. A production system consists of a set of workstations and a transfer mechanism to handle the part flows. Workstations are parallel configurations of machines that perform specific types of operations. Due to deterioration, machines are prone to failures that occur at random instants. A failed machine remains inoperable for some period of random duration until it is repaired and ready to operate again. Various types of raw parts are successively loaded to machines and are finally transformed into finished products. Due to machine breakdowns and different processing rates, the part flow between adjacent workstations may be reduced or interrupted. To avoid this, buffers of finite capacity are placed in front of each workstation to provide temporary storage space for workparts. Optimal design and control of production systems are problems of major economic importance in modern industries and depend on the development of efficient analytic tools to evaluate the performance of candidate optimization strategies. Analysis is required in designing a new system, in expanding an existing one, and in short-term production control, with the objective of achieving a desired performance while keeping investment and operational costs low. Performance measures, such as average throughput rates and product cycle times, are estimated by exploiting the model of the system. This contribution provides an in-depth treatment of techniques for modeling and the development of optimal control policies for production networks. Because of the great importance of this issue in manufacturing sytems, this is a particularly appropriate contribution with which to begin this volume.

The next contribution is "Scheduling, Optimization, and Control in Automated Systems," by S. David Wu and Richard A. Wysk. The performance of an automated manufacturing system not only relies on state-of-the-art machinery but also on the effective planning and control of day-to-day operations. Of critical importance is the scheduling of production operations and the control of planned activities. Traditional methods to shop scheduling typically involve the generation of a static, precomputed schedule using optimization or heuristic methods. The static schedule is generated based on certain managerial or operational objectives subject to shop constraints. In real-world applications, however, the moment a schedule is released to the shop, various uncertainty aspects of the system will render this schedule nothing but a far-reaching guideline for shop operations. As a result, the planning of activities in the shop relies heavily on human intervention. This contribution is an in-depth treatment of the broadly complex and interacting issues of scheduling, optimization, and control in production systems.

The next contribution is "Performability of Automated Manufacturing Systems," by N. Viswanadham, Y. Narahari, and R. Ram. Automated manufacturing systems (AMSs) can be regarded as flexible, degradable, fault-tolerant systems. To evaluate such systems, combined measures of performance and reliability, called performability measures, are needed. The most important performability measures in the AMS context are related to throughput and manufacturing lead time (MLT), since high productivity and low lead times are prime features determining the competitiveness of AMSs. Performability has a strong relation to the notion of flexibility in manufacturing systems. Performability modeling is an active research topic in the area of fault-tolerant computing systems. Performability studies are of great interest in the AMS context also, since performability enables one to quantify flexibility and competitiveness of a manufacturing system. Steady-state performability, interval performability, and distribution of performability with respect to throughput and MLT measures are considered. Through several illustrative examples and numerical results, the importance of performability modeling and evaluation in AMS design is brought out.

The next contribution is "Modeling, Control, and Performance Analysis of Automated Manufacturing Systems Using Petri Nets," by Frank Dicesare and Alan A. Desrochers. The utilization of Petri nets has become more important in recent years because it can solve problems that cannot be modeled using queueing theory, while avoiding the time-consuming, trial-and-error approach of simulation. The modeling problem is characterized by concurrent and asynchronous events that are typical for such discrete event dynamic systems. Petri nets are well suited for modeling manufacturing systems because they capture the precedence relations and interactions among these events. In addition, a strong mathematical foundation exists for describing these nets. This allows a qualitative analysis of such system properties as deadlock, conflict, and boundedness. The Petri net model can also be used as the basis of a real-time controller for a manufacturing system. The flow of tokens through the net establishes the sequence of events to carry out a specific task, such as the manufacturing of a particular part type. Petri net controllers have been used in factories in Japan and Europe. Because of the power of the utilization of Petri net techniques in manufacturing systems, this contribution is an essential element of this set of volumes.

The next contribution is "*Evalcon*: A System for Evaluation of Concurrency in Engineering Design," by Andrew Kusiak and Edward Szczerbicki. Design undergoes evaluation at all of its stages. Evaluation of design from the perspective of various life-cycle attributes, such as manufacturability, assemblability, reliability, and so on, is named a concurrency evaluation. The concurrency evaluation process in routine design should assist a designer in specification, selection, and synthesis of parts. The main difficulty associated with the evaluation process is caused by the incompleteness and uncertainty of information. To facilitate the routine design and to allow for incremental concurrency evaluation as the design evolves, interactive design support techniques are presented in this contribution, and their essential importance in the product design process in manufacturing systems is illustrated by examples that are presented in this contribution.

The next contribution is "Concurrent Engineering: Design of Assemblies for Schedulability," by Andrew Kusiak. The traditional design of assemblies (products) has relied on an iterative approach. The main difficulty with the iterative approach is that it is time consuming and many iterations are required before a design project is completed. In order to reduce the design cycle, concurrent design has been introduced. The basic idea of concurrent design is to shorten the time horizon in which the design constraints, such as schedulability, manufacturability, quality, reliability, maintainability, and so on, are introduced. This contribution focuses on the schedulability constraint. It is shown that assembly design has a significant impact on the complexity of the scheduling problem and its solution quality.

The next contribution is "Multi-Criteria Optimization and Dynamic Control Methods in Flexible Manufacturing Systems," by In-Kyu Ro and Joong-In Kim. There are many methods used to solve scheduling and operational control problems in flexible manufacturing systems (FMSs). These methods are mathematical programming techniques (optimization and heuristic methods), queueing networks, computer simulation, Petri nets, and expert systems, etc. Most FMS models are based on one out of the several methods listed above and they are single-objective oriented techniques. However, the combination of several methods may give better results than a single one and decision makers usually consider multiple objectives. On the other hand, most FMS models consider the scheduling for jobs and automated guided vehicles (AGVs) separately. However, jobs and AGVs are closely related, and thus their simultaneous optimal scheduling is essential in manufacturing systems. This contribution presents techniques for these important issues and illustrates them with numerous examples.

The next contribution is "Computer-Aided Manufacturing Systems Design: Framework and Tools," by Bin Wu. In this contribution Wu presents a framework of computer-aided manufacturing systems design (CAMSD) and discusses the tools and methodologies that can be used in an integrated environment. Two distinct approaches can be taken to system design, i.e., the top-down and the bottom-up. The former starts with a set of objectives and then creates a system model that fits the intended purpose. This approach gives little consideration to the current system being operated. Although preferable from the designer's point of view, it can result in a design that requires a total replacement of the current system and, therefore, heavy capital investment. The other approach bases its consideration mostly on the existing system, producing designs that require less capital investment. However, since it is unlikely that any manufacturing unit was initially designed to cope with later structural and functional alterations, the options available could be constrained and ideas severely limited. The approach presented in this contribution is a hybrid of these two approaches. Dr. Wu is one of the leading research workers in manufacturing systems on the international scene, and he has written one of the most important texts on this subject (*Fundamentals of Manufacturing Systems Design and Analysis*, Chapman and Hall, London, 1991). Therefore, his contribution on the greatly important subject of

CAMSD constitutes a most welcome element of this series of volumes.

The next contribution is "Software Configuration Techniques in Operational Systems," by M. Sloman, J. Kramer, J. Magee, and P. Butryn. There has been considerable research into programming languages that emphasize modularity and the reuse of components (e.g., ADA, SR, Argus, NIL, and Emerald). However, there has been comparatively little work on languages that emphasize the separation of programming individual components (programming in the small) from the specification and construction of distributed systems from predefined, reusable components (programming in the large). The various languages mentioned above do have some support for combining separately compiled modules, but they treat the resultant system as a single large program. The specification of a current system is buried within the current state of a program, which makes it difficult to modify or extend the system. The diversity of the software components useful in large distributed systems, such as manufacturing systems, implies that appropriate, state-of-the-art programming languages should be used. For instance, high-level procedural languages (e.g., C, Pascal, and Modula2) should be used for the real-time control and monitoring, object-oriented languages (e.g., Smalltalk and C++) for man–machine interfaces, artificial intelligence languages (e.g., Prolog and Lisp) for the expert systems and knowledge bases needed for the advanced intelligent (i.e., decision-making) components, and Fortran for the existing numerical analysis packages. Although the rewards are great, the use of heterogeneous programming languages exacerbates the problems of integration, as they may make use of incompatible communication mechanisms and data representations. This contribution presents a configuration approach to integrating heterogeneous software components that communicate and interact in a distributed computer automation system.

The next contribution is "Software for Dynamic Tool Modeling in Manufacturing and Automation," by Luis C. Cattani and Paul J. Eagle. The number of robot applications has increased dramatically since 1980 and the predominant application has been spot welding in the automotive industry. Unfortunately, spot welding tools or guns typically impart large loading on robots and have made robot reliability an important issue. Mechanical failures have been identified with the symptoms of excessive backlash and harmonic drive failures. Excessive backlash can cause poor weld quality while the harmonic drive failures can cause downtime. In spot welding applications, robot overloading is caused by the weld guns producing large reaction torques on the robot, exceeding the robot design capacity. An often-overlooked source of robot loading occurs due to dynamic effects. This is due to the difficulty of calculating the moments of inertia of the welding guns. This complex geometry is a result of the sizes and shapes of the welding tools needed to operate on car bodies. To address this problem, a simple solid modeling software called TANGO-UD has been developed by Cattani and Eagle, and it is presented in this contribution. It calculates and analyzes solid properties (mass, volume, moments of inertia, products of inertia, and radius of gyration) of welding guns and other mechanical components. The software has a geometric and material definition language, a parser and compiler, a mass properties calculator, a mass properties analyzer, and the ability to display the wire frame model of the object being analyzed using the AutoCAD graphics software.

The next contribution is "Techniques for Optimal Operation Allocation Methods in Manufacturing Systems—A Review of Non-Probabilistic Approaches," by J. M. Wilson. The field of optimal operation allocation methods has been a rich source for theoretical research and applications development in the manufacturing systems subject area. Starting from the early work of A. S. Manne in the 1950s, the subject of optimal allocation has moved forward with great rapidity in the 1960s and 1970s. Developments after the 1970s might have seemed to be entering into a rather arid phase, with more research than applications to use it. However, two major changes took place that put new momentum into research and applications, and this momentum has been maintained into the 1990s and is continuing. The two developments were (1) the classification of the complexity of algorithms for many types of problem and (2) the introduction of flexible manufacturing systems. This contribution provides an in-depth treatment of optimal operation allocation techniques.

The final contribution to this volume is "A Branch-and-Bound Algorithm for Solving the Machine Allocation Problem," by Chunhung Cheng, Andrew Kusiak, and Warren J. Boe. Group technology (GT) is concerned with the formation of part families and machine cells. The result of grouping machines and parts may lead to either a physical or logical machine layout. The physical machine layout implies rearrangement of machines on the shop floor. The logical machine layout requires virtual grouping of machines and therefore does not alter the position of machines on a shop floor. In this contribution an in-depth treatment of the problem of allocating machines to cells (machine layout) in an optimum manner is presented, and, because of the importance of this issue in manufacturing systems, it is an appropriate contribution with which to conclude this volume.

This volume is a particularly appropriate one as the third of a companion set of five volumes on techniques and technologies in manufacturing and automation systems. The authors are all to be congratulated for their superb contributions, which will provide a uniquely significant reference source for workers on the international scene for years to come.

TECHNIQUES IN MODELING AND CONTROL POLICIES FOR PRODUCTION NETWORKS

YANNIS A. PHILLIS
VASSILIS S. KOUIKOGLOU

Department of Production Engineering & Management
Technical University of Crete
Chania 731 00, Greece

I. INTRODUCTION

Large, interconnected production systems are widely used in manufacturing. A production system consists of a set of workstations and a transfer mechanism to handle the part flows. Workstations are parallel configurations of machines which perform specific types of operations. Due to deterioration, machines are prone to failures which occur at random instants. A failed machine remains inoperable for some period of random duration until it is repaired and ready to operate again.

Various types of raw parts are successively loaded to machines and are finally transformed into finished products. Due to machine breakdowns and different processing rates, the part flow between adjacent workstations may be reduced or interrupted. To avoid this, buffers of finite capacity are placed in front of each workstation providing temporary storage space for workparts.

Optimal design and control of production systems are problems of major economic importance in modern industries and depend on the development of efficient analytic tools to evaluate the performance of candidate optimization strategies. Analysis is required in designing a new system, expanding an existing one, and in short-term production control, with the objective of achieving a desired performance while keeping investment and operational costs low. Performance measures, such as average throughput rates and product cycle times, are estimated by exploiting the model of the system.

Once a computationally efficient and accurate estimation method is developed, it can be used to evaluate alternative designs and control policies.

CONTROL AND DYNAMIC SYSTEMS, VOL. 47

Typical decision parameters are the following:

a) Number of parallel machines in workstations;
b) Production rates of machines;
c) Number of repairmen in the system;
d) Repair allocation policies for systems with limited number of repairmen;
e) Buffer capacities.

The problem then becomes one of determining the decision parameters to maximize the profit subject to capital investment and physical limitations.

The problems of analysis and optimization have been studied by many researchers. Yet, they remain in general unsolved, due to the inherent complexity of production systems and the lack of an exploitable model for their description.

Markovian models for queueing systems have been used to analyze the steady-state characteristics of production systems. These models were developed to study various stochastic processes arising in computer, communication, and traffic networks. At any instant of the production period the system state is determined by the number of parts in buffers and the operational conditions of machines. Closed form solutions for the steady-state probabilities exist for networks with infinite storage capacities and reliable machines having exponential processing times [1-4]. For production lines with unreliable machines and zero or infinite storage spaces it is possible to calculate the throughput rate in terms of the stand-alone throughput rates of machines [5,6].

Systems with limited storage are difficult to study due to the enormity of the state vector. However, two-stage systems with an intermediate buffer have been efficiently analyzed in [7-12]. The basic assumptions adopted therein include phase-type processing times or unreliable machines with constant or exponential processing times, continuous or discrete-part flow, and possible scrapping of parts. By exploiting the difference equations of the corresponding Markov model the steady-state probabilities are found to have a sum-of-products form.

Larger systems can be approximated using decomposition and aggregation techniques [13-20]. The idea is analogous to the Norton equivalent in electrical networks. The system is divided into two subsystems, one being upstream and the other downstream a buffer. Each subsystem is represented by an equivalent machine. Hence the whole system is approximated by a simple two-stage one. The operating characteristics of each equivalent machine are determined by carrying out further decompositions of the corresponding subsystem.

Another approximate method is the mean value analysis which was developed for the study of computer networks [21,22]. The solution procedure is based on the result that various performance measures of a queueing system can be expressed in terms of average buffer levels and waiting times in queues. Little's formula is an analogous result for two-stage systems. More general production networks can be efficiently approximated [23].

A minimax algebra model proposed recently in [24] has been applied to deterministic systems. The method takes into account the timing constraints which rule the order of part arrivals and departures from machines. The main advantage of this approach is that it can be applied to both transient and steady states. Yet only a few results have been reported concerning unreliable production networks [25].

While these methods require small computation times, they are based on rather restrictive assumptions such as exponential-type distributions for processing, failure, and repair times of machines and steady-state operation. In addition, most of them suffer from the curse of dimensionality. In fact, the state vector consists of the probabilities of all possible states which may be visited by the system and its dimension grows exponentially with system size. For a completely general analysis of production systems, simulation is the inevitable alternative.

A simulation model describes a possible evolution of the production process. Such models known also as piece-by-piece or brute-force, use a computer program to observe the workparts as they move sequentially through the production system. The system state consists partly of discrete states, such as buffer levels and machine operating conditions, and continuous states, such as machine residual times-to-failure, repair, or production of a workpart. When a continuous-state variable reaches zero, one or more discrete states are modified. In a piece-by-piece simulation model the discrete-state transitions, arrivals or departures from machines are marked as elementary events. To each machine and buffer a candidate event is assigned along with the corresponding time of occurrence. The smallest event time determines the next state transition. After executing a next event, the state vector is modified according to a set of rules imposed by the functional characteristics of the production system. Using a random number generator together with appropriate distribution functions, the times of occurrence of new events are computed. The process then repeats.

This technique is called discrete-event simulation because it views the production system as a discrete event dynamic system (DEDS). In the last two decades a number of general purpose simulation languages have been developed such as SLAM, SIMSCRIPT and GPSS, which are widely used in modeling DEDS's.

Various researchers have also used simulation in optimization. Early studies that appeared in the literature examine the problem of buffer capacity allocation and control of production lines [26-29]. However, a common disadvantage of simulation versus analytical models is their long run times. There are two reasons behind this, the time consuming piece-by-piece tracking of the system function and the need for several independent runs for obtaining the desired accuracies for various candidate designs. Efforts towards combining analytic and simulation models have achieved considerable reduction in computation times. In [30] a unifying definition and classification of such hybrid models is given.

Perturbation analysis (PA) is a hybrid technique that calculates the partial derivatives of performance measures of a DEDS with respect to decision parameters. By using PA during a simulation run one can predict the evolution of a perturbed system resulting from a small change in a parameter without needing additional simulation. This is accomplished by applying a number of simple perturbation rules. The corresponding computational savings versus brute-force experimentation are proportional to the number of decision parameters. The method has been applied together with gradient optimization procedures in allocation of buffer capacities for production lines and machine average production rates for queueing networks [31-34]. Extensions to PA proposed in [35,36] yield accurate sensitivity estimates with respect to discrete-decision parameters in a number of experiments.

Recently, a new hybrid event-driven method has been developed which increases the execution speed of a simulation run. Its efficiency over piece-by-piece methods results from the reduced number of events that are observed by the simulator, namely a machine fails or recovers, and a buffer fills or empties. The system evolution between successive events is tracked analytically. Such a model is distinguished as a Class II hybrid model in [30], since it uses simulation and analysis interactively. In [37,38] continuous-flow systems are modeled, whereas discrete-part production lines are examined in [39]. An integrated buffer design algorithm developed in [40] combines PA with the hybrid model to study continuous production lines. Finally in [41], the problem of repair allocation is studied for discrete production lines.

In this article we develop a state space model for production networks based on the discrete-event approach presented in [38,39]. The model describes accurately production lines in both transient and steady-state operations, and is approximate but quite fast for multiproduct networks. The corresponding algorithms are then used to solve two optimization problems. The first concerns the optimal allocation of repair resources and buffer space in production lines . The problem is to maximize the average profit subject to operational constraints. The optimal solution is obtained using a steepest ascent procedure; the gradient information is extracted from simulation experiments by applying a set of simple PA rules. The second problem involves optimal control of production networks with limited number of repairmen. Here we use the model to compare the performance of various state-independent control policies.

II. PRODUCTION LINES

A. SYSTEM DESCRIPTION

A production line is a series arrangement of machines and intermediate buffers, (see Fig.1). Parts enter the first machine and are processed and transported downstream, until they finally leave the system. Machines produce at deterministic but not necessarily equal rates and fail and are repaired randomly. The system is maintained by a number of repairmen. When a machine breaks down a repairman is sent to it immediately. If all repairmen are busy the machine remains inoperable and waits until a repairman is available.

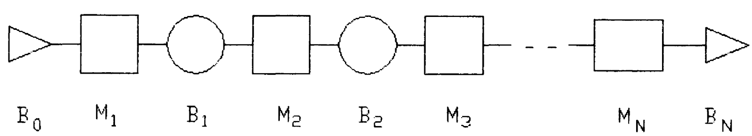

Fig. 1. A production line with N machines.

The operation of the production line is ruled by the following:

(1) The line consists of N machines and N+1 buffers. There is one buffer B_0 at the beginning of the line with finite capacity and another B_N at the end with unlimited capacity.

(2) The nominal production rate of each machine M_i is deterministic. Uptimes and repair times are assumed, for convenience, to be exponential random variables although any type of distribution may be considered. The system is maintained by Q repairmen, where $Q \leq N$.

(3) In each machine there is space for a single workpart. A machine M_i is starved if it has no part to work on and the inventory of the upstream buffer B_{i-1} has been exhausted. Also M_i is blocked if it is prevented from releasing a finished part downstream because B_i is full. In the

sequel the terms empty or starved and full or blocked will be used equivalently.

(4) Starved or blocked machines remain forced down until a workpart or a unit space is available. During these idle periods machines do not deteriorate.

(5) Transportation time of workparts to and from buffers is negligible or is incorporated in the processing time.

It is crucial in our simulation model that production rates be deterministic and known. The assumption that idle machines are not subject to failures has been verified for most production systems [42]. This implies that a machine breaks down only after an amount of production. Time-dependent failures caused by exogenous factors, such as power supply failures, may be easily taken into account by using appropriate random numbers which mark the time of occurence of these phenomena.

The model proposed is event-driven. There are three buffer states: empty, intermediate (partially full), and full. There are three machine states: up, down, and under repair, that is, a repairman is working on a broken machine.

State transitions of machines or buffers will be referred to as **discrete events** or simply **events**.

Now we define the system parameters:

NR_i: nominal production rate (workparts/time-unit) of machine M_i, $i=1,\ldots,N$.

$1/p_i$: mean time-to-failure of M_i operating at nominal rate.

$1/r_i$: mean time-to-repair of M_i.

BC_j: capacity of buffer B_j, $j=0,1,\ldots,N$.

For any $k = 1,2,\ldots$, let t_k be the time and e_k the type of the k-th event of the system. The states of the system components (machines and buffers) are evaluated after the k-th event occurence. At time t_k, right after the occurence of e_k the state of the production line is described by:

$$M_i(k) = \begin{cases} \text{U, if machine } M_i \text{ is up} \\ \text{D, if } M_i \text{ is down} \\ \text{R, if } M_i \text{ is under repair} \end{cases}$$

$$B_j(k) = \begin{cases} \text{E, if buffer } B_j \text{ is empty and } M_{j+1} \text{ is starved} \\ \text{F, if } B_j \text{ is full and } M_j \text{ is blocked} \\ \text{I, otherwise an intermediate state.} \end{cases}$$

The above discrete states remain unchanged in the intervals $[t_k, t_{k+1})$ between successive events. We write $B_j(k) = B_j(k+1^-)$ and $M_i(k) = M_i(k+1^-)$.

Next event indicators $\varepsilon_k(c)$ and the associated next-event times $\tau_k(c)$ mark those events that will take place in every component c of the line at time t_k. The next event $\varepsilon_k(M_i)$ for machine M_i belongs in the set {machine-fails, machine-begins-repair, machine-completes-repair}; the next event $\varepsilon_k(B_j)$ for buffer B_j belongs in the set {buffer-empties, buffer-fills, buffer-becomes-not-full, buffer-becomes-not-empty}. Next events and event times are functions of the discrete states. Discrete states, however, do not provide adequate information to predict the evolution of the system. For example, consider a buffer in intermediate state. Its level increases or decreases as workparts arrive or depart until a buffer-fills or buffer-empties event takes place. To compute the type and time of next event we must know the frequencies of part arrivals and departures which in turn are tied to the operations of the upstream and downstream machines.

The dynamic system we study has a more frequently varying section which determines its short-term behavior. The corresponding states will be henceforth referred to as continuous states, in constrast to the slow varying discrete states. At time t_k the continuous state vector consists of:

$BL_j(k)$	level of buffer B_j
$a_j(k)$	residual time-to-next arrival to B_j
$d_j(k)$	residual time-to-next departure from B_j
$TP_i(k)$	total production of machine M_i
$PF_i(k)$	residual parts-to-failure of M_i
$TR_i(k)$	residual time-to-repair of M_i,

for $i = 1, 2, ..., N$ and $j = 0, 1, ..., N$. In the sequel we drop k whenever this is convenient. The variables $PF_i(k)$ and $TR_i(k)$ are nonincreasing with time. When PF_i equals zero and machine M_i receives an additional workpart, then a breakdown takes place. Similarly when TR_i equals zero the machine resumes operation instantly. In our model these variables are the outputs of appropriate random number generators.

B. SUMMARY OF THE METHOD

A simulation experiment provides a possible evolution of the physical system during a period of operation. We propose a hybrid simulation-analytic approach to the analysis of the DEDS herein which uses simulation and analysis interactively. The algorithm avoids the excessive computational burden of piece-by-piece models by updating the system state only at times when an event occurs, rather than at every part arrival or departure from buffers. Keys to this reduction are the deterministic production rates of machines and the localized effects of event occurrences in the system states.

We call $R_i(k)$ the rate of machine M_i which is a discrete state variable updated after k-th event occurs according to the following rules (see Fig. 2):

1) If a machine is up and it is neither starved nor blocked, then it produces at its nominal rate. The production cycle is the inverse of the production rate.
2) The production cycle of a starved machine M_{i+1} is the time between successive arrivals. Its production rate is set equal to the minimum of the rates NR_{i+1} and R_i.
3) The production cycle of a blocked machine M_i is the time between two successive departures and similarly its rate is set equal to the minimum of the rates of NR_i and R_{i+1}.

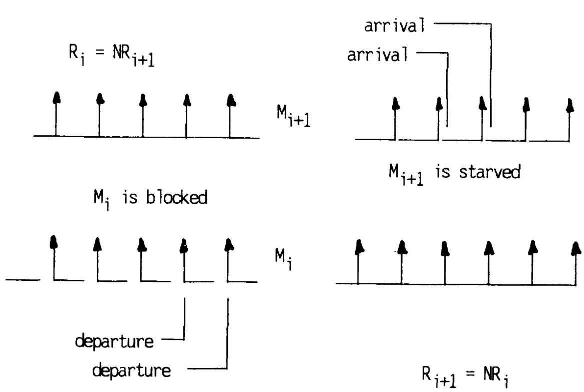

Fig. 2. Effects of events to production rates.