INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Time dependent models of material flow in small-lot assembly systems

Wang, Li-Chih, Ph.D.
The Ohio State University, 1989

Copyright ©1989 by Wang, Li-Chih. All rights reserved.
TIME DEPENDENT MODELS OF MATERIAL FLOW
IN SMALL-LOT ASSEMBLY SYSTEMS

DISSERTATION
Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of the Ohio State University
By

Li-Chih Wang, B.S., M.S.

* * * * *

The Ohio State University
1989

Dissertation Committee:
Dr. Wilbert E. Wilhelm
Dr. Albert B. Bishop
Dr. Haikady N. Nagaraja

Approved by

Adviser
Department of Industrial and Systems Engineering
To My Parents, Wife and Son
ACKNOWLEDGEMENTS

I deeply appreciate my adviser, Dr. Wilbert E. Wilhelm for his invaluable help, encouragement, patience and guidance throughout my Ph.D. program.

I express sincere appreciation to other members of my dissertation committee, Drs. Albert B. Bishop and Haikady N. Nagaraja for their valuable suggestions and comments. I also wish to thank Dr. Gordon M. Clark, Dr. Ronald L. Lewis and Dr. Dik Lee for serving on my graduate study committee.

Finally, I wish to express my gratitude to my parents and wife for their support, understanding and encouragement throughout my graduate study. Their faith in my abilities has allowed me to reach my goals and set my sights even higher.

This research was supported by the National Science Foundation under Grant Numbers DMC-8500898 and DMC-8896303.
VITA

June 6, 1959 ..................................... Born: Taipei, Taiwan, R.O.C.

1981 .................................................... B. S. in Industrial Engineering, Tunghai University, Taiwan, R.O.C.

1985 ..................................................... Graduate Research Associate, Dept. of Ind. & Sys. Eng., The Ohio State University, Columbus, Ohio.

1985 ..................................................... M. S. in Industrial Engineering, The Ohio State University, Columbus, Ohio.

1986 - 1989 .......................................... Graduate Research Associate, Dept. of Ind. & Sys. Eng., The Ohio State University, Columbus, Ohio.

PUBLICATIONS


FIELDS OF STUDY

Studies in Applied Operations Research in Manufacturing Systems

Dr. Wilbert E. Wilhelm, Dept. of Industrial and Systems Engineering.
Dr. Albert B. Bishop, Dept. of Industrial and Systems Engineering.
Dr. Gordon M. Clark, Dept. of Industrial and Systems Engineering.

Studies in Computer Aided Manufacturing

Dr. Ronald L. Lewis, Dept. of Industrial and Systems Engineering.

Studies in Data Base

Dr. Dik Lee, Dept. of Computer and Information Science.
TABLE OF CONTENTS

ACKNOWLEDGMENTS .............................................................................. iii
VITA ............................................................................................................... iv
LIST OF TABLES ........................................................................................... x
LIST OF FIGURES ......................................................................................... xii

CHAPTER PAGE

I. INTRODUCTION ................................................................... 1
  1.1 Background ................................................................... 1
    1.1.1 Cellular Small-Lot Assembly Systems ....... 2
    1.1.2 Binary Assembly Network ....................... 5
  1.2 Research Objectives ..................................................... 8
  1.3 The Approach ............................................................... 11
    1.3.1 Robotic Cells .................................................... 11
    1.3.2 Binary Assembly Network ....................... 12
  1.4 Overview of the Dissertation .................................... 14

II. LITERATURE REVIEW .................................................................. 15
  2.1 Introduction ...................................................................... 15
  2.2 Models of Robotic Cells .............................................. 15
  2.3 Prior Research on Recursion Models ....................... 18
  2.4 Models of PERT Networks ........................................... 20
  2.5 Phase-Type (PH) Distributions .................................. 23
  2.6 Tchebycheff System (T-System) and Its Applications .................................................. 25
  2.7 Summary of the Literature Review ....................... 30
5.1 Introduction ................................................................. 73
5.2 Model 2: Approximation of the Distribution Functions of Operation Finishing Times .......... 76
  5.2.1 Moments of the Maximum of Two Independent Random Variables and the Convolution with Another Random Variable ............................................................. 77
  5.2.2 Approximation of the Distribution Function of a Nonnegative Random Variable by a Discrete Distribution ................................................................. 79
    5.2.2.1 Lower Principal Representation ...... 82
    5.2.2.2 Upper Principal Representation ..... 83
  5.2.3 Approximation of the Distribution Function of a Nonnegative Random Variable by a Mixture of Erlang Distributions ................................. 84
    5.2.3.1 Lower Principal Representation .... 86
    5.2.3.2 Upper Principal Representation for a Random Variable with support on [0,B] ................................................ 87
    5.2.3.3 Upper Principal Representation for a Random Variable with support on [A,B] ................................................ 89
  5.2.4 Approximation of the Distribution Function of Operation Finishing Time by a Mixture of Erlang Distributions ................................................ 91
5.3 Model 3: Bounds on the Expected Values of Operation Starting and Finishing Times .......... 93
  5.3.1 New Lower Bounds .................................................... 93
  5.3.2 Devroye's Lower and Upper Bounds .......... 97
<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Configuration for Evaluating Recursion Model Accuracy</td>
<td>56</td>
</tr>
<tr>
<td>4.2 The Effect of Lmax for Different Job Routings</td>
<td>59</td>
</tr>
<tr>
<td>4.3 Comparison of Estimates for Different Job Routings</td>
<td>60</td>
</tr>
<tr>
<td>4.4 System Design for I/O Buffer Analysis</td>
<td>63</td>
</tr>
<tr>
<td>4.5 The Effect of I/O Buffers on System Performance</td>
<td>64</td>
</tr>
<tr>
<td>4.6 System Design for Industrial Setting</td>
<td>68</td>
</tr>
<tr>
<td>4.7 Expected Robot Service Times</td>
<td>68</td>
</tr>
<tr>
<td>4.8 CCA Family Composition and Mix</td>
<td>69</td>
</tr>
<tr>
<td>4.9 Expected Setup and Insertion Times</td>
<td>69</td>
</tr>
<tr>
<td>4.10 Comparison of Estimates in an Industrial Setting</td>
<td>70</td>
</tr>
<tr>
<td>5.1 Number and Locations of Mass Points for the Lower and Upper pr's for Given First Three Moments</td>
<td>81</td>
</tr>
<tr>
<td>6.1 Values of $\chi$ and $p_1$ at Two Different Levels</td>
<td>120</td>
</tr>
<tr>
<td>6.2 Moments of Operation Finishing Times of Nodes in a Four-level, Balanced BAN with Case 1 Settings</td>
<td>121</td>
</tr>
<tr>
<td>6.3 Parameters of the df of Network Completion Time Approximated by The Lower and Upper pr's in Test #1 (a)</td>
<td>124</td>
</tr>
</tbody>
</table>
6.4 Moments of Operation Finishing Times of Nodes in an
Eight-level, Balanced BAN with Case 1 Settings (\(\chi=25, p_1=0.4\)) .... 132

6.5 Parameters of the df of Network Completion Time
Approximated by The Lower and Upper pr's in Test #1 (b) .......... 135

6.6 Moments of Network Completion Time with Case 2 Settings .... 142

6.7 Distribution Function of Network Completion Time
in Test #2 ......................................................................................... 143

6.8 The Means and Variances of Operation Finishing Times of
Nodes in an Eight-level, Balanced BAN with Case 3 Settings ...... 149

6.9 The Means and Variances of Operation Finishing Times of
Nodes in an Eight-level, Imbalanced BAN with Case 3 Settings .. 151
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Feasible Configuration of a Robotic Cell</td>
</tr>
<tr>
<td>1.2</td>
<td>An L-level Assembly Tree</td>
</tr>
<tr>
<td>3.1</td>
<td>Correlation Relationships</td>
</tr>
<tr>
<td>3.2</td>
<td>Notation for Network</td>
</tr>
<tr>
<td>4.1</td>
<td>Hypothetical Robotic Cell</td>
</tr>
<tr>
<td>4.2</td>
<td>Relative Difference Between Recursion and Simulation Estimates of Expected Activity Beginning Time vs Activity Number (Hypothetical Robotic Cell)</td>
</tr>
<tr>
<td>4.3</td>
<td>System Configuration of an Industrial Setting</td>
</tr>
<tr>
<td>4.4</td>
<td>Relative Difference Between Recursion and Simulation Estimates of Expected Activity Beginning Time vs Activity Number (Industrial Setting)</td>
</tr>
<tr>
<td>5.1</td>
<td>A Four-level Binary Assembly Network (BAN)</td>
</tr>
<tr>
<td>6.1</td>
<td>PDF's of the Lower and Upper (A=0.1) pr's with ((c-1/c, \gamma)=(-1.175, .415))</td>
</tr>
<tr>
<td>6.2</td>
<td>PDF's of the Lower and Upper (A=0.0) pr's with ((c-1/c, \gamma)=(-1.175, .415))</td>
</tr>
<tr>
<td>6.3</td>
<td>PDF's of Weibull Distribution Compared with Johnson and Taaffe's Approximation with ((c-1/c, \gamma)=(-4.967, -.373))</td>
</tr>
</tbody>
</table>
6.4 PDF's of the Lower and Upper (A=0.0) pr's with 
(c-1/c, γ)=(-4.967, -.373) ................................................................. 109

6.5 PDF's of Weibull Distribution Compared with Johnson and 
Taaffe's Approximation with (c-1/c, γ)=(-2.284, .206) ...................... 110

6.6 PDF's of the Lower and Upper (A=0.0) pr's with 
(c-1/c, γ)=(-2.284, .206) ................................................................. 111

6.7 PDF's of Weibull Distribution Compared with Johnson and 
Taaffe's Approximation with (c-1/c, γ)=(-.978, .917) ................... 112

6.8 PDF's of the Lower and Upper (A=0.0) pr's with 
(c-1/c, γ)=(-.978, .917) ................................................................. 113

6.9 PDF's of Gamma Distribution Compared with Johnson and 
Taaffe's Approximation with (c-1/c, γ)=(-1.658, .940) .................... 115

6.10 PDF's of the Lower and Upper (A=0.0) pr's with 
(c-1/c, γ)=(-1.658, .940) ................................................................. 116

6.11 PDF's of Gamma Distribution Compared with Johnson and 
Taaffe's Approximation with (c-1/c, γ)=(1.0, 3.236) ...................... 117

6.12 PDF's of the Lower and Upper (A=0.1) pr's with 
(c-1/c, γ)=(1.0, 3.236) ................................................................. 118

6.13 PDF of Network Completion Time Approximated by 
the Lower and Upper pr's in Run 1 of Test #1 (a) ....................... 125

6.14 PDF of Network Completion Time Approximated by 
the Lower and Upper pr's in Run 2 of Test #1 (a) ....................... 126

6.15 DF of Network Completion Time Approximated by
the Lower and Upper pr's in Run 1 of Test #1 (a) .......................... 128

6.16 DF of Network Completion Time Approximated by
the Lower and Upper pr's in Run 2 of Test #1 (a) .......................... 129

6.17 PDF of Network Completion Time Approximated by
the Lower and Upper pr's in Run 1 of Test #1 (b) .......................... 136

6.18 PDF of Network Completion Time Approximated by
the Lower and Upper pr's in Run 2 of Test #1 (b) .......................... 137

6.19 A Three-level BAN Consisting of Dummy Activities ............... 139

6.20 The Single Source/Single Sink PERT Network which
Corresponds to Figure 6.19 .............................................................. 139

6.21 The Order-of-Processing (OP) Diagram which
Corresponds to Figure 6.20 .............................................................. 140

6.22 DF of Network Completion Time Obtained from Kulkarni
and Adlakha's Backward Algorithm in Test #2 .......................... 144

6.23 DF of Network Completion Time Approximated by
the Lower and Upper pr's in Test #2 ................................................ 145

6.24 PDF of Network Completion Time Approximated by
the Lower and Upper pr's in Test #2 ................................................ 147

6.25 PDF of Network Completion Time Approximated by
the Lower and Upper pr's in Test #3 (Balanced Scenario) ............. 154

6.26 PDF of Network Completion Time Approximated by
the Lower and Upper pr's in Test #3 (Imbalanced Scenario) ....... 155
CHAPTER 1

INTRODUCTION

1.1 Background

Recent emphasis on methods to assure just-in-time operations in small-lot assembly systems underscores the importance of carefully managing material flow to ensure the coordination of assembly operations so that an effective control on work-in-process inventory and the assembly schedule can be obtained [Saboo and Wilhelm (1986)]. Basically, material flow may be defined as the transformation process from raw material to final product. Hence, material flow planning concerns analysis of the time involved in the acquisition of parts (or raw materials) and the transformation of parts (or subassemblies) to complete product(s) according to predetermined management policies.

In small-lot (batch) production systems, assembly operations present a major challenge to material flow management, since they require coordinated delivery of parts (subassemblies) for subassembly (assembly). Inadequate coordination may increase work-in-process and prevent assembly according to schedule [Wilhelm et al (1986)]. As a result, the cycle time required to assemble a batch may increase, delaying the release time of subsequent batches. Hence, the overall operating cost will increase.

Due to time-varying production requirements, part shortages, and random disruptions (e.g., machine or material handling equipment
breakdown), assembly systems rarely operate under steady state conditions. Thus, the study of transient performance (i.e., time dependent material flow) of small-lot assembly systems is of much practical importance.

There are two primary purposes of this dissertation. One is to model the transient performance of a small-lot assembly system which is organized in cellular configurations; the other is to model time dependent operations in a binary assembly network (BAN). It is to be noted that cellular production/assembly systems and BAN represent two important configurations in modern production/assembly systems. The former may include a broad variety of features found in most types of production systems (e.g., generalized flowshop or jobshop with material handling and finite buffer capacities), the latter represents the manufacture of a product in an assembly system in which materials are assembled according to the relationships indicated by a BAN.

1.1.1 Cellular Small-Lot Assembly Systems

The reason for organizing an assembly system in a cellular configuration is to achieve the objectives of reducing in-process inventory, tooling changes, material handling and cycle times and of simplifying production planning and control problems. Cellular manufacturing is the physical division of manufacturing machinery into production units; each cell is designed to produce a specific part family. A part family is a set of parts that require similar machinery: tooling, machining operations, jigs
Manufacturing cells may be classified according to the following three classes [Greene (1987)]:

1. Operators set up the machine and perform a portion of the process.

2. Operators set up the machine, the machine or transfer line is automatic, and operators load/unload parts.

3. The machine, set up and transfer line are automatic, and material handling equipments load/unload parts.

In this dissertation, automated material handling equipments (e.g., robots) are considered as the devices to transport materials between/within cells and to load/unload machines. The overall "system" may consist of several different functional cells, each composed of a set of stations. Each station may consist of more than one machine, may have an input (I) buffer that serves as a queue for lots which are waiting for service and an output (O) buffer that can receive processed parts. The capacity of I/O buffers must be determined by the system designer. The cell itself also has I/O buffers to allow exchange of materials with other cells. Raw material (finished products) may be stored in the input (output) buffer of the system. Figure 1.1 illustrates one possible system configuration in which either flow-shop or job-shop routing of lots is allowed within a cell.

To date, most research has concentrated on evaluating cell design, but few models have been developed to evaluate cell performance according to different operating policies (e.g., lot routing, lot sequencing rules) [see Section 2.1 for review]. Analytical models of the transient performance of even a simple assembly system (e.g., a two
Figure 1.1 Feasible Configuration of a Robotic Cell.
station assembly line with no buffers) are difficult to obtain once the assumptions of Poisson input and exponential service times are violated [Saboo (1986)]. The first primary purpose of this dissertation is to develop an approximation model that can evaluate both alternative designs and operating policies in a cellular, small-lot assembly system. In particular, a recursion procedure (see Chapter 3) is developed to efficiently and accurately model transient material flows in cellular, small-lot assembly systems.

Currently, recursion models are based on the assumption that processing (or operation) times in assembly systems are normally distributed, which is a reasonable assumption for small-lot assembly systems under certain conditions (e.g., lot size is greater than 10 or small lot size with many operations). The assumption of normally distributed operation time becomes weaker when both lot size as well as the number of operations are small (e.g., lot size = 1, no rework) [see Wilhelm (1987), Seidmann et al (1987) and Lee (1985) for justifications]. It is, thus, interesting and important to study time dependent operations in an assembly system and to provide a means for material flow managers to plan and control an effective assembly schedule when the operation times are not normally distributed [Wilhelm (1988)].

1.1.2 Binary Assembly Network

In general, material (flow) relationships in an assembly may be represented by an assembly tree, a graph in which nodes represent stations
that perform (sub)assembly operations and arcs indicate the flow of materials (e.g., parts, subassemblies, semi-finished products) from one station to the next. In an assembly tree, each node has at most one successor and may have more than one predecessor; predecessors represent components that are assembled at the node (i.e., station) [Wilhelm (1988)]. The operation at any node may start only if all required components from its immediate predecessors are available. An L-level assembly tree in which nodes in level L represent parts (or raw materials) required for assembly of the final product, nodes in level (L-m) (m=1,2,...,L-2) denote subassemblies (or components) required by nodes in level (L-m-1) [node one represents the final product] is depicted in Figure 1.2.

By evaluating operation starting and finishing times, estimated performance may be compared with scheduled performance. That is, material flow through the assembly process may be evaluated. With the above information, material flow managers may plan and control an effective assembly schedule in a stochastic assembly environment. For example, parameters of the distributions of operation times could be set to define an effective assembly schedule, which indicates when components (or subassemblies) required by each subassembly (assembly) should be ready. Appropriate control tactics can be evaluated to achieve the due date of the product determined by the master production schedule. For example, these tactics may reschedule the predetermined due date for components ordered from outside vendors and for the final product that a
Figure 1.2: An L-Level Assembly Tree
customer ordered if some disruptions (e.g., machine breakdown) occur in the dynamic assembly environment.

Since any assembly tree may be represented as an equivalent binary assembly network (BAN), in which each node has exactly one immediate successor and two immediate predecessors (i.e., two components are assembled in each operation) [Saboo (1986) and Wilhelm (1988)]. Furthermore, it is known that the distribution of any nonnegative random variable may, in general, be arbitrarily closely approximated by a mixture of Erlangs. The second primary purpose of this dissertation is to develop models to evaluate the operation starting and finishing times of nodes in a BAN with the assumption that processing (or operation) times are distributed as mixtures of Erlangs.

1.2 Research Objectives

In this dissertation, a recursion procedure is developed to model material flow in a cellular, small-lot assembly system (i.e., a set of interacting robotic cells), and models are developed to describe time dependent operations in an assembly system in which relationships among assembly operations are represented by a binary assembly network.

Specific objectives of this research are:

(1) Development of a computationally efficient recursion procedure to model the transient performance (i.e., material flow) in cellular,
(2) Incorporation of important and realistic features of cell operations (e.g., material handling, job-shop routing, finite I/O buffers, lot/machine sequencing, parallel machines etc.) in the recursion model.

(3) Development of models to approximate the distributions of operation finishing times of nodes in a BAN with the assumption that processing times are distributed as mixtures of Erlangs.

(4) Development of a new lower bound on the expected values of operation starting and finishing times of nodes in a BAN with the assumption that processing times are distributed as mixtures of Erlangs.

Since most of the computation time involved in implementing the recursion procedure is required to estimate the correlations between the beginning times of different (robot) activities, the recursion procedure presented in this dissertation develops a new approach to efficiently estimate correlations so that objective (1) can be achieved.

Prior research on recursion models (see Section 2.3 for review) cannot be applied to a complicated manufacturing system, such as a robotic cell, with the variety of features listed above. Achieving objective (2) would make recursion models more widely applicable in actual applications, so they could be used as aids in improving the productivity of cellular, small-lot production/assembly systems.
Currently, recursion models must invoke the assumption that operation times (e.g., machine processing and material handling times) are normally distributed. There is no analytical or approximation method available to describe the time dependent material flow in an assembly system when processing (or operation) times are distributed according to some general distributions. Hence, achieving objectives (3) and (4) is important and practical.

Studying operation starting and finishing times of nodes in a BAN will provide information about the start and finish times of subassembly, assembly and final product operations in an assembly facility. The practical significance of achieving objective (3) is that it provides a basis for the material flow managers to plan and control an effective assembly schedule and to coordinate the progress of assembly operations to meet the scheduled performance when processing (or operation) times are distributed as mixtures of Erlangs [Wilhelm (1988)].

The importance of achieving objective (4) (i.e., a lower bound) can also be recognized by the fact that it provides conservative and less optimistic estimates of the expected values of operation starting and finishing times compared to those obtained from deterministic methods, which simply consider expected operation times and provide an overly optimistic estimate of system performance that cannot be achieved in a stochastic assembly environment.
1.3 **The Approach**

This section describes the basic concepts of the recursion procedure, which is applied to model a robotic cell. Also, concepts of modeling starting and finishing times of operations in a BAN are described.

1.3.1 **Robotic Cells**

The recursion procedure, denoted Model 1, developed for the robotic cell is based on the idea that an activity (i.e., robot operation) can begin when all required resources (e.g., robot, lot, machine) are ready. That is, the activity beginning time may be defined as the greatest of those resource ready times. After the activity is initiated, the ready times of those resources may be determined by appropriately adding robot service and/or machine processing times to the activity beginning time. Hence, recursive relationships exist among activity beginning and resource ready times. It is assumed that robot service and machine processing times are (normally distributed) random variables, activity beginning and resource ready times are, thus, also random variables with correlations caused by the recursive relationships.

Therefore, evaluating the distribution of activity beginning time is actually the same as evaluating the distribution of the maximum of a set of dependent random variables. Since no mathematical theory which allows the maximum to be described exactly, approximating the distribution of the maximum becomes a viable alternative. The recursion
model approximates the joint distribution of activity beginning and resource ready times as a multivariate normal with the assumption that robot service and machine processing times are mutually independent and normally distributed and applies Clark's (1961) equations (see Section 3.2.3.1 for details) to approximate the moments of activity beginning time. The method involves estimating the pairwise correlations among resource ready times.

In a robotic cell, the correlations between pairs of resource ready times may be estimated from the correlations between the corresponding (robot) activity beginning times and associated beta factors (as explained in Section 3.3.1). In order to efficiently estimate the correlations between certain activity beginning times, a "fill-in on the fly" procedure, which works backwards and computes only necessary correlations, is developed. Due to the lack of exact description of transient material flow in a robotic cell, simulation estimates are used to evaluate the accuracy of the recursion model developed in this dissertation.

1.3.2 Binary Assembly Network

With the assumptions that all components are ready at time zero and that processing times are distributed as mixtures of Erlangs, the exact moments of operation finishing times in the (L-1)th level may be obtained. According to the Tchebycheff system (T-system) introduced by Karlin and Studden (1966), the distribution of any nonnegative random variable (rv) may be approximated by matching its moments with those of
a discrete rv. Applying the relationships between the moments of a discrete rv and one which is distributed as a mixture of Erlangs, the distribution of any nonnegative rv may be approximated by matching its moments with those of a mixture of Erlang distributions. Hence, the distributions of operation finishing times in the \((L-m)^{th}\) \((m=1,\ldots,L-1)\) level may be approximated by mixtures of Erlang distributions.

The idea of this approximation method is to evaluate the moments of operation starting times in the \((L-m)^{th}\) \((m=2,\ldots,L-2)\) level, which may be obtained by evaluating the maximum of two independent rv's [i.e., operation finishing times for nodes in the \((L-m+1)^{th}\) level] which are approximated by mixtures of Erlangs. The moments of operation finishing times may then be evaluated by forming the convolution of operation starting and processing times. Hence, the distributions of operation finishing times for nodes in the \((L-m)^{th}\) level are approximated by mixtures of Erlangs, and the moments (distributions) of operation starting (finishing) times of nodes in the \((L-m-1)^{th}\) \((m=2,\ldots,L-2)\) level may be evaluated by applying the concept described above (see Model 2 in Section 5.2 for details).

Based on the derivation of the lower bound on the expected value of the maximum of two independent random variables, each of which is the convolution of a nonnegative rv with known expected value and a rv which is distributed as a mixture of Erlangs, a new lower bound on the expected values of operation starting times for nodes in the \((L-m)^{th}\) \((m=1,\ldots,L-1)\) level may be obtained. The lower bound on the expected operation finishing time is simply the addition of the lower bound on the
expected operation starting time and the corresponding mean processing time (see Model 3 in Section 5.3 for details).

In order to evaluate the accuracy of approximation methods developed in this dissertation, the recursion model is evaluated relative to a number of different system features and in a hypothetical industrial setting. Approximation methods for the distributions of operation starting and finishing times in a BAN are evaluated in numerical tests with different types of processing time distributions.

1.4 Overview of the Dissertation

In Chapter II, prior research on robotic cells and recursion models is reviewed, and existing models of PERT networks and phase-type (PH) distributions are discussed. Also, the Tchebycheff system and its applications are described in details. In Chapter III, the development of the recursion model (Model 1) and the computational procedure used to estimate correlations between resource ready times are described. The model is evaluated relative to different system features and also in an hypothetical industrial setting in Chapter IV. In Chapter V, Model 2 (approximation of the distribution functions of operation finishing times) and Model 3 (bounds on the expected values of operation starting and finishing times) for a BAN are developed. Numerical evaluation of Models 2 and 3 is discussed in Chapter VI. Finally, a summary, conclusions and recommendations for future research are given in Chapter VII.
CHAPTER II

LITERATURE REVIEW

2.1 Introduction

In this dissertation, a recursion procedure (Model 1) is developed to model material flow in a cellular, small-lot assembly system. In Section 2.2, models for robotic cells are reviewed. Previous research on recursion models is described in Section 2.3. Models 2 and 3 are developed to evaluate the operation starting and finishing times of nodes in a BAN, which may be interpreted as a PERT network. Existing models of PERT networks are, therefore, reviewed in Section 2.4. In Models 2 and 3, processing times are assumed to be distributed as mixtures of Erlangs. Phase-type (PH) distributions are briefly reviewed in Section 2.5, since mixtures of Erlangs belong to this more general class of distributions. Properties of Tchebycheff systems, which play an important role in Model 2, are described in Section 2.6.

2.2 Models for Robotic Cells

A number of researchers have modeled flexible manufacturing systems (FMS), including Wilhelm and Sarin (1983), Van Looveren et al (1986), Seidmann et al (1986) and Buzacott and Yao (1986). All of these papers provide comprehensive literature reviews on models of FMS.
Compared to FMS, models for robotic cells have not been so well developed. In general, models of robotic cells can be categorized as: addressing cell design or cell operation. This research emphasizes material flow, an aspect of the cell operation phase.

Sarin and Wilhelm (1983) and Wilhelm and Sarin (1985) reviewed models available for the design of robotized manufacturing cells; the main issues that they discussed were the layout of the cell and sequencing the movements of the robot to effectively tend machines. Suri (1985) reviewed a number of quantitative techniques [e.g., deterministic (probabilistic)/steady state (transient) techniques] for analysis of robotic systems in each of the two phases.

Seidmann and Nof (1985) developed a capacity model that incorporates the influence of stochastic feedback flow on the productivity of a unitary manufacturing cell. A probabilistic analysis of part flow with product recirculation for rework is presented. Cases involving batch production and interleaving several product types are developed by Seidmann et al (1985).

Cutkosky et al (1983) discussed the conceptual design of a manufacturing cell for small batch production. Their design philosophy is to partition cell components and control programs into modules according to function. Medeiros and Sadowski (1982) developed a general modeling approach for the design and analysis of computer-controlled robotic cells. Their approach allows a simulation model to be constructed by selecting appropriate modules from an available library and linking them together. Simulation models are probably most popular means of analyzing robotic
cells due to their capability of studying both design operational problems. Hansen (1983), Steudel (1986), Biles and Bathina (1986) and Badalamenti and Bao (1986) developed simulation models which address some of the steps involved in the design and analysis of robotic cells.

One of the main objectives of this research is to model material flow in a robotic cell. To date, many commercial simulation packages have been developed to provide a detailed analysis of robotic cells; many packages focus primarily on the graphical illustration of robot operations. Emphasis has not been placed on the analysis of material flow using robots. Most of the research which addresses material flow problems in robotic cells utilizes simulation models to study system operation.

Robinson and Nof (1983, 1986) developed a generalized robotic cell simulator for examining both cell layout and flow control problems. The benefit of this simulator is that users can specify rules to effect decisions. Nof and Drezner (1986) discussed the robotic assembly plan problem and provided a solution procedure to optimize robotic assembly by prescribing component part feeding and pick-insert sequencing. Grant and Wilson (1986) proposed use of a general-purpose simulation language to evaluate the flow of materials through robotic manufacturing systems. Their research focuses on production issues and the relationship of robotic components to the rest of the system; an example is provided to illustrate how the various components may be linked together for overall evaluation of system performance. Chu and Moodie (1986) experiment with an adaptive control methodology for material handling systems containing robots and conveyors. The control strategy which integrates the
functions of simulation, optimization, database, pattern recognition and robot control attempts to increase the flexibility of automated material handling equipment in a specific computer integrated manufacturing (CIM) system.

Wilhelm (1986a) applied the recursion approach in a paper which is, perhaps, most closely related to this research, since it models transient material flow in robotized manufacturing cells. Some assumptions invoked in that paper (e.g., each workpiece must be processed on one and only one machine in the cell and the robot manipulator must return to a 'null' position upon completing each loading or unloading operation) are relaxed by this research.

2.3 **Prior Research on Recursion Models**

Recursion models were first tested by Wilhelm and Ahmadi-Marandi (1982); they developed a recursive procedure to describe operating characteristics of assembly systems and indicated that the transient performance of assembly systems may be modeled by estimating operation starting times, which can be represented as the maximum of a set of resource available (ready) times. The main research problem is then focused on estimating the correlations among resource available times. Recursion models have been applied to several production configurations, such as the assembly line [Wilhelm et al (1986)], assembly network [Saboo and Wilhelm (1984,1986)], and flowshop [Wilhelm (1986)]. Wilhelm and Wang (1986) also examined the accuracy of applying Clark's (1961)
approximation procedure to describe the maximum of dependent normal random variables in a part kitting application.

Earlier recursion models are summarized by Saboo (1986); he developed different computational procedures to estimate correlations and, hence, permit modeling the transient operations of three types of systems - the assembly line, assembly network and generalized flowshop. However, the algorithms of earlier recursion models, including Saboo's, are rather limited and cannot be applied to complicated systems such as robotic cells due to the following reasons: (1) earlier recursion models process jobs on a station-by-station basis and, thus, cannot model features such as job-shop routing; (2) a large number of correlations must be estimated whenever an operation is finished. Hence, a number of redundant correlations are computed, causing computational inefficiencies. There are some other limitations; for example, (1) the computational procedures for estimating correlations are system dependent (different types of systems require different computational procedures) and (2) the transportation time between stations is not considered.

Wilhelm, Kalkunte, and Cash (1987) developed recursion models of robotic assembly cells. Their approach treats a small number of resources in the cell using a resource-based model of operations, which incorporates the logic of GERT and extends the fundamental concepts of earlier recursion models. Although their approach did increase the flexibility of recursion models somewhat, it still cannot be applied to the robotic cell considered in this dissertation due to the following reasons: (1) when the
system is complicated (e.g., the system which may consist of several cells, each cell having several stations, each station having a certain number of machines and input/output buffers), it is not efficient to compute and revise correlations between the ready times of each pair of resources whenever an activity is initiated as their method requires; (2) their computational algorithm is suitable for a system with "fixed" job routing (i.e., the operation sequence was largely determined by precedence relationships), but not good in a system with complex job routing, such as the robotic cell with job-shop routing.

2.4 Models of PERT Networks

PERT, which has been widely applied in project management, is concerned with a 'project' comprised of a large number of 'activities' that are arranged in a complex 'network' [Hartley and Wortham (1966)]. A number of studies have been motivated to evaluate the underlying statistical distribution of the completion time, $T_N$, of a PERT network (when the distributions of individual activity durations are known).

In order to illustrate the difficulty of evaluating the distribution function (df) of $T_N$, let $\Omega$ be the set of all paths in the activity network and $F_j$ be the duration of path $j \in \Omega$. Then, $T_N$ is determined by the duration of the longest path [Dodin (1985b)]:

$$T_N = \max_{j \in \Omega} \{ F_j \}$$

Hence, for $t \geq 0$, 
\[ H(t) = \Pr(T_N \leq t) = \Pr(\max_{j \in \Omega} F_j \leq t) \]

In general, it is extremely difficult to evaluate \( H(t) \), for \( t \geq 0 \), due to the interdependence of \( F_j, j \in \Omega \). Although the special structure of a binary assembly network (BAN), which is described in Chapter 5, may eliminate certain interdependencies among \( F_j, j \in \Omega \), the mathematical equation of \( H(t) \) is still quite complicated when the distribution of individual activity durations is a mixture of Erlang distributions.

Some models, [e.g., Fisher and Goldstein (1983) and Kulkarni and Adlakha (1986)], have been developed to model the distribution function of project completion time in a PERT network with mutually independent and exponentially distributed activity durations. Although these models could be extended to treat Erlang-distributed activity durations, the state space would grow rapidly, making the calculation of \( H(t) \) intractable when the number of nodes or the order of the Erlang distributions increases. Due to the difficulty of obtaining exact forms for \( H(t) \), a number of authors have considered developing bounds on \( H(t) \) as an alternative. The inequalities which are usually used to obtain bounds are [Shogan (1977)]:

\[
\Pr(\max_{j \in \Omega} F_j \leq t) \leq \min_{j \in \Omega} \{\Pr(F_j \leq t)\} \quad (2.1)
\]

\[
\Pr(\max_{j \in \Omega} F_j \leq t) \geq \prod_{j \in \Omega} \Pr(F_j \leq t) \quad (2.2)
\]

\[
\geq \max \left\{ 0, \left[ \sum_{j \in \Omega} \Pr(F_j \leq t) \right] - (m - 1) \right\} \quad (2.3)
\]
in which, \( m \) is the number of paths in \( \Omega \). It is to be noted that inequality (2.2) holds only when random variables \( F_j (j \in \Omega) \), with random vector \( F=(F_1,F_2,...,F_J) \), are associated [i.e., \( \text{Cov}[g_1(F),g_2(F)] \geq 0 \) for all nondecreasing functions \( g_1(\cdot) \) and \( g_2(\cdot) \)].

Kleindorfer (1971) and Shogan (1977) use inequalities (2.1) and (2.2) to obtain bounds on \( H(t) \) for a PERT network, but their procedures cannot be applied to the BAN in which activity times are distributed as mixtures of Erlangs because their methods require the assumption that activity times are discrete random variables. Furthermore, the difficulty of performing convolution and multiplication of random variables whose df's are mixtures of Erlang distributions prevents application of the reduction and duplication processes introduced by Dodin (1985b) to obtain a lower bound on \( H(t) \).

While bounds on the expected completion time have been developed by a number of researchers, approaches developed by Fulkerson (1962), Elmaghraby (1967) and Robillard and Trahan (1976) can only handle discrete activity durations. Devroye (1979) developed an approach to obtain upper bounds on the mean and variance of completion time for a PERT network using only the means and variances of individual activity durations. Aven (1985) developed a new procedure to obtain the upper bound on the mean of the maximum of a number of random variables. It can be shown that the upper bound obtained from Aven's procedure is tighter than that from Devroye's method. A comprehensive review of bounds on the expected value of the maximum of a number of random
variables could be found in Arnold (1988). Robillard and Trahan (1977) use inequality (2.3) to approximate the df of project completion time in a PERT network with discrete activity durations. Dodin (1985a) developed an approximation model for a PERT network with discrete activity durations or with continuous activity durations which can be 'discretized', the accuracy of his approximation method depends heavily on the degree of error in discretizing the continuous distributions.

2.5 **Phase-Type Distributions**

The phase concept was first introduced by Erlang (1917), who studied the distribution, which is the convolution of a finite number of independent, identically distributed exponential random variables, and is now known as the Erlang distribution. A k-stage Erlang distribution with rate $\lambda$ can be interpreted as the distribution of the time required to successively traverse through k states (or stages, phases), where the time spent in each state is exponentially distributed with rate $\lambda$. The Erlang distribution can be used to represent exponential, normal and degenerate distributions by varying the value of k. One limitation of the Erlang distribution is that its squared coefficient of variation is always between zero and one.

Cox (1955) showed that any distribution on $[0, \infty)$ whose Laplace transform is rational may be represented by a sequence of exponentially distributed phases, now known as the Coxian distribution. For a k-stage Coxian distribution, the sojourn time of the $i^{th}$ ($i=1,2,...,k$) phase, if visited,
is exponential with rate $\lambda_i$; the probability of entering the $i^{th}$ ($i=1,2,...,k$) stage from the $(i-1)^{st}$ stage is $p_{i-1}$ and the probability of exiting the set of phases from the $(i-1)^{st}$ ($i=1,2,...,k$) stage is $1-p_{i-1}$. A $k$-stage Erlang distribution is a special case of the Coxian distribution (i.e., with $p_{i-1}=1$, $\lambda_i=\lambda$, for $i=1,2,...,k$).

For some distributions with a rational Laplace transform, the corresponding (approximating) Coxian distribution may have complex parameters or a large number of phases. This would cause computational difficulty. Altiok (1985a) developed a set of formulas for a two-stage coxian (phase-type) distribution to approximate any general distribution with known squared coefficient of variation greater than 1. Altiok (1985b) also modeled production lines with phase-type operation and repair times.

Neuts (1981) introduced a phase-type (PH) distribution which subsumes all of the generalizations mentioned above. A nonnegative random variable $X$ is said to be of phase-type (PH) if $X$ can be represented as the time until absorption in a finite-state, continuous-time Markov chain. In other words, this is the distribution of the sojourn time in a network of independent phases, each with exponential sojourn time, where Markov routing occurs between phases and the initial stage is determined by an initial probability vector [Johnson (1987)]. A comprehensive review of PH distributions can be found in Neuts (1981; Chapter 2). PH distributions exhibit some important properties: the set is closed under convolution as well as under finite mixtures. But the number of parameters required to specify a unique PH distribution makes the fitting of a PH distribution a very difficult task.
Johnson and Taaffe (1988a) state that a mixture of Erlang distributions provides a basis for several assertions about the versatility of PH families. By choosing \( r \) large enough, any degenerate distribution can be arbitrarily closely approximated by an \( r \)-stage Erlang distribution. Hence, by mixing sufficiently many Erlang distributions, each with sufficiently high order, any continuous distribution on \([0, \infty)\) can be arbitrarily closely approximated by a mixture of Erlang distributions due to the fact that there exists a sequence of discrete distributions that converges in distribution to this continuous distribution [Johnson and Taaffe (1988a)]. A special case of the mixture of Erlang distributions is the hyperexponential distribution, which is a finite mixture of exponential distributions with different rate parameters. The hyperexponential distribution is also capable of approximating any distribution on the nonnegative reals with squared coefficient of variation greater than or equal to one [Johnson (1987)].

2.6 **Tchebycheff System and Its Applications**

A Tchebycheff system (T-system) consists of a set of continuous functions \( \{u_i(t)\}_{i=0}^{n} \) defined on a real interval \([a, b]\), for which every nontrivial real linear combination \( \sum_{i=0}^{n} a_i u_i(t) \), with \( \sum_{i=0}^{n} a_i^2 > 0 \), has at most \( n \) distinct zeros in \([a, b]\). That is, functions \( \{u_i(t)\}_{i=0}^{n} \) are called a T-system over \([a, b]\) if the \((n+1)\)st order determinants [Karlin and Studden (1966)]
Central to the T-system is the moment space, $M_{n+1}$, (with respect to $\{u_i(t)\}_{i=0}^n$) which is generated in the following way:

$$U \left( t_0, t_1, \ldots, t_n \right) = \begin{vmatrix} u_0(t_0) & u_0(t_1) & \cdots & u_0(t_n) \\ u_1(t_0) & u_1(t_1) & \cdots & u_1(t_n) \\ \vdots & \vdots & \ddots & \vdots \\ u_n(t_0) & u_n(t_1) & \cdots & u_n(t_n) \end{vmatrix} > 0$$

in which $a \leq t_0 < t_1 < \ldots < t_n \leq b$.

A function $f$ is said to be of bounded variation if there is a constant $\alpha$ such that for an arbitrary partition: $a=t_0 < t_1 < \ldots < t_m=b$, we have the inequality [Sprecher (1970)]:

$$\sum_{k=1}^{m} |f(t_k) - f(t_{k-1})| \leq \alpha$$

In the context of probability theory, $\sigma(t)$ may be interpreted as a distribution function. If $u_i(t)=t^i$, then $c_i$ is the $i$th moment of $\sigma(t)$, and $M_{n+1}$ is the set of all combinations of feasible $0^{th}$, $1^{st}$, ..., $n^{th}$ moments [Johnson (1987)]. If $c$ is a boundary point of $M_{n+1}$, moments $c_i$ ($i=0,1,\ldots,n$) uniquely determine a random variable (rv), $T$, which has moments $c_i$ ($i=0,1,\ldots,n$).
Actually, $T$ is a discrete rv with $(n+2)/2$ or fewer mass points on the interval $[a, b]$ [see Karlin and Studden (1966); Theorem II.2.1]. Karlin and Studden (1966) also introduced the lower, $T_L$, and upper, $T_U$, principal representations (pr's) of $c$, $c=(c_0, c_1, ..., c_n)$, for which $c$ is an interior point of $M_{n+1}$ [see Karlin and Studden (1966); Chapter II.3, II.6]. That is, if we consider one more continuous function $u_{n+1}(t)$, and

$$c_{n+1} = \int_a^b u_{n+1}(t) \, d\sigma(t) \quad (\sigma \in V(c)) ,$$

(2.6)

$$V(c) = \left\{ \sigma \mid c_i = \int_a^b u_i(t) \, d\sigma(t), i = 0, 1, ..., n \right\} ,$$

in which $V(c)$ is the set of all feasible distribution functions of a random variable which can represent $c$. If $\{u_0(t), u_1(t), ..., u_n(t), u_{n+1}(t)\}$ still form a $T$-system [i.e., satisfy equation (2.4)], then rvs $T_L$ and $T_U$ will have the same $c_i$'s ($i=0,1, ..., n$), but correspond to the lower and upper bounds on $c_{n+1}$, respectively, as shown by Karlin and Studden (1966).

For several different types of partial characterizations (e.g., certain moments) of a nonnegative random variable, Eckberg (1977) derived sharp upper and lower bounds on the Laplace-Stieltjes Transform (LST) of the corresponding distribution function of this random variable (in terms of those given partial characterizations) according to the properties of Tchebycheff systems introduced by Karlin and Studden (1966). The characteristics of those derived extremal distributions are that they are discrete distributions with mass at only two or three points. In latter
sections, it will be shown that these discrete distributions could be fitted by mixtures of Erlang distributions.

By considering only the first two moments of a nonnegative random variable (rv), Whitt (1984b) extended Eckberg's result, deriving formulas to obtain both lower and upper principal representations, which are mixtures of exponential distributions. Since only the first two moments are considered and higher ordered moments are ignored, the shape of the density function (pdf) of the lower and upper pr's may be totally different from the original one. Another restriction of Whitt's approach is that the squared coefficient of variation (cv) of a hyperexponential distribution is always greater than or equal to one, so it is impossible to approximate any random variable with cv^2 < 1. In fact, Whitt (1984a) also indicated that the third moment is important when the cv of the approximated distribution is large (i.e., cv > 1).

Johnson and Taaffe (1987, 1988a) investigate the use of a mixture of Erlang distributions of common order for matching moments to the distribution of a nonnegative random variable, X. Let the coefficients of variation and skewness of the distribution of X be denoted by c and γ, respectively. Johnson and Taaffe show that, except for the case in which γ=c-1/c, the first n moments of the distribution of any nonnegative random variable may be matched by a mixture of \([n/2]+1\) r-stage Erlang distributions, for sufficiently large r. (The notation \([n/2]\) denotes the greatest integer less than or equal to n/2). Johnson and Taaffe derived a three moment matching procedure which is actually the lower principal representation (pr) of a T-system introduced by Karlin and Studden (1966).
Since the concepts which underly their derivation are the same as those related to one of the two approximation methods developed in this dissertation, formulas of determining parameters of the approximation of the df of rv $X$ derived by Johnson and Taaffe (1988a) are briefly reviewed below. Differences between Johnson and Taaffe's method and the new approaches developed in this dissertation are noted in Chapter 5.

Theorem 4 in Johnson and Taaffe (1988a) gives formulas for approximating the distribution function, $G$, of a nonnegative random variable, $X$, as a mixture of two Erlang distributions by matching its first three moments. Let $m_1$, $m_2$ and $m_3$ be the first three noncentral moments of $X$ and $G_i$ ($i=1,2$) be an $r$-stage Erlang distribution with rate parameter $\lambda_i$ ($i=1,2$). The mixing probability of the $i$th branch of $G$, $G_i$, is $p_i$ ($i=1,2$) so that $G(t) = \sum_{i=1,2} p_i G_i(t)$. Since the formula for determining $r$, the common order of $G_i$ ($i=1,2$), will also be used in the new approximation methods developed in this dissertation, it is presented as a lemma in Chapter 5. Formulas for determining parameters $\lambda_1$, $\lambda_2$ and $p_1$ ($p_2 = 1-p_1$) derived by Johnson and Taaffe (1988a) are listed as follows:

$$\lambda_i^{-1} = \left(-Y \pm \sqrt{Y^2 - 4XZ}\right) / (2X)$$ \hspace{1cm} (2.7)

$$p_1 = \left(\frac{m_1}{r} - \lambda_2^{-1}\right) / \left(\lambda_1^{-1} - \lambda_2^{-1}\right),$$ \hspace{1cm} (2.8)

in which
\[ X = r(r + 2)m_y, \]
\[ Y = -\left( r \times + \frac{r(r + 2)}{r + 1} y^2 + (r + 2)m_2^2 y \right), \]
\[ Z = m_1 x, \]
\[ y = m_2 - \left( \frac{r + 1}{r} \right)m_2^2, \quad \text{and} \]
\[ x = m_1 m_3 - \left( \frac{r + 2}{r + 1} \right)m_2^2. \]

2.7 **Summary of the Literature Review**

In summary, a number of researchers have used simulation models to design cells and to study the operating characteristics of production and assembly cells. Existing commercial simulation packages focus primarily on graphical illustration of the robot operations but not on the analysis of material flow. Earlier recursion models have been applied to assembly flowlines, assembly networks, generalized flowlines and robotic handling in a simple cell. This dissertation presents an efficient, new method for applying a recursion procedure in cellular, small-lot assembly systems which incorporate a comprehensive set of features, including those studied in earlier recursion models as well as others (e.g., job shop routing, lot/machine sequencing, and material handling between and within cells) that cannot be treated by previous methods.

Research related to PERT networks can only model (approximate) the distribution of network completion time in which activity times are either exponentially distributed or discrete random variables. Existing models provide either bounds on the distribution and moments of network completion time for a PERT network with discrete activity
durations or bounds on the mean and variance of network completion time based on the means and variances of individual activity durations. This dissertation develops models to evaluate the distribution and moments of operation starting and finishing times for nodes in a binary assembly network in which activity times are distributed as mixtures of Erlangs.
CHAPTER III

MODEL FOR ROBOTIC CELLS (MODEL 1)

3.1 Introduction

The objective of this chapter is to present a computationally efficient recursion procedure to model the transient performance (i.e., time dependent material flow) in cellular, small-lot assembly systems. In Section 3.2, the fundamental relationships which are required to approximate the transient behavior of robotic cells are described. A new computational procedure, which may efficiently estimate pairwise correlations among resource ready times, is also presented. Finally, in Section 3.3, logic for implementing the recursion procedure is described.

3.2 Model Development

In a robotic cell, robots are assumed to play the role of material handling and/or machine tending devices; therefore, each operation (or activity) is initiated and terminated by a robot. For example, a machine cannot begin an operation unless a lot, a set of workpieces, is loaded by a robot. A lot will remain on the machine (even if it is completed) until it is unloaded by a robot. Therefore, if we can keep track of the relationships among robot activities, the transient performance of a robotic cell may be described.
Although the sequence of stations visited by each lot, \( \{j | j=1,2,...,J\} \), is determined by the lot routing, a lot may or may not actually visit an input buffer (I), and an output buffer (O) at each station. Visits to I/O buffers depend upon cell operations over time and cannot be predetermined. Therefore, we may treat buffers and machines as locations \( \{l | l=1,2,...,L\} \); each move starts with a lot which is processed and waiting to be unloaded from some location \( \{c | c=1,2,...,L\} \) and ends when the lot is loaded at its destination location \( \{d | d=1,2,...,L\} \) (c ≠ d).

### 3.2.1 Model Assumptions

Assumptions made for modeling the robotic cell are:

1. All machine processing times and material handler (e.g., robot) service times are normally distributed and mutually, statistically independent of each other.

2. Cell configuration is known.

3. Lot routings and lot/machine sequencing rules are known.

### 3.2.2 Fundamental Relationships in Robotic Cells

In a robotic cell, a robot activity cannot be initiated unless three conditions are satisfied: (1) the required robot is available, (2) the lot is ready for this move (e.g., a lot was processed at its upstream station and is waiting for a robot to transport it to its next station), and (3) a destination is ready to receive that lot. When robot \( r \) becomes available to perform
another activity, its current location and the number of the last activity it completed, n-1, are known. If lot j seeks robot r to move it from its current location c to its destination d, which is also seeking a lot, then, the beginning time of the n th activity of robot r, B cdjnr, may be stated mathematically as:

\[ B_{cdjnr} = \max [ R_{nr}, A_{cj}, D_{dj} ] \]  \hspace{1cm} (3.1)

in which,

\[ R_{nr} = \text{time at which robot r is ready to initiate its n th activity.} \]
\[ A_{cj} = \text{time at which lot j is ready to move from location c.} \]
\[ D_{dj} = \text{time at which destination d is ready to receive lot j.} \]

The times at which location c becomes available to receive its next lot ( \( D_{c,j^+} \) (j + indicates the next lot that will be loaded at location c), robot r is ready to perform its next activity ( \( R_{n+1,r} \)) and lot j is completed at destination d and waiting for the next movement ( \( A_{dj} \)) are determined by:

\[ D_{c,j^+} = B_{cdjnr} + G_{cjnr} \]  \hspace{1cm} (3.2)
\[ R_{n+1,r} = B_{cdjnr} + G_{cjnr} + T_{cdjnr} \]  \hspace{1cm} (3.3)
\[ A_{dj} = B_{cdjnr} + G_{cjnr} + T_{cdjnr} + P_{dj} \]  \hspace{1cm} (3.4)

in which,

\[ G_{cjnr} = \text{time required for robot r to travel from its current location to location c, grasp lot j and unload it from c.} \]
\[ T_{cdjnr} = \text{time required for robot r to transport lot j from location c to destination d, load it at d and release it.} \]
\[ P_{dj} = \text{processing time of lot } j \text{ at destination } d. \]

Robot service times are defined by \( G \) and \( T \) variables; \( P \) is the processing time if \( d \) is a machine and is zero if \( d \) is a buffer. It is assumed that random variables \( B, G, T \) and \( P \) on the right hand side of each equation (3.2) - (3.4) are mutually independent, the means and variances of \( D, R \) and \( A \) may, thus, be evaluated by taking the appropriate additions. The procedure for evaluating activity beginning time, \( B \), defined in equation (3.1) is described in Section 3.2.3.

3.2.2.1 Performance Measures

A variety of performance measures may be derived using the expected values of activity beginning and resource ready times:

1. \( E[M_d] = \text{Expected makespan for station } d \)
   \[ = E[D_{dj}] - E[B_{d*^n_r*r^*}] \]
   in which,
   \( E[D_{dj}] \) = expected completion time of the last operation performed by station \( d \) (on lot \( J_d \))
   \( E[B_{d*^n_r*r^*}] \) = expected starting time of the first activity at station \( d \) (on lot \( j^* \))

2. \( C_k = \text{Expected cycle time for a lot of type } k \)
   \[ = \sum_{j=1}^{N_k} (E[A_{d*^j}] - E[A_{j^1}]) / N_k \]
in which,

\[ E[A_{d,j}] = \text{expected completion time for a lot, } j, \text{ of type } k \text{ at } d^* \]
(the last station on its routing)

\[ E[A_{1j}] = \text{expected launch time for a lot, } j, \text{ of type } k \text{ at the first station on its routing} \]

\[ N_k = \text{total number of type } k \text{ lots.} \]

(3) \[ U_d = \text{Expected utilization of station } d \]

\[ = \sum_{j=1}^{J_d} \frac{E[P_{dj}]}{E[M_d]} \]

in which,

\[ E[P_{dj}] = \text{expected processing time of lot } j \text{ processed at station } d. \]

(4) \[ U'_d = \text{Expected utilization of machines at station } d \]

\[ = U_d / N'_d \]

in which,

\[ N'_d = \text{the number of machines at station } d. \]

(5) \[ U''_r = \text{Expected utilization of robot } r \]

\[ = \text{the sum of the expected times for all activities performed by robot } r, \text{ divided by the expected makespan of robot } r \]

\[ = \sum_{n'=1}^{n_r} \frac{E[G_{c_{jn}'r}] + E[T_{c_{d_jn}'r}]}{E[R_{n_r+1,r}] - E[B_{c^*d^*jr}]} \]

in which,

\[ n_r = \text{the total number of activities performed by robot } r. \]

(6) \[ E[WIP] = \text{Expected work-in-process inventory over } J \text{ lots} \]
\[
E[A_{j^*}] = \sum_{j=1}^{J} \frac{E[A_{j^*}]}{\max_{d} E[M_d]}
\]

in which,

\(E[A_{j^*}]\) = expected time that lot \(j\) leaves the system output buffer.

\(J\) = the total number of lots.

\(\max_{d} E[M_d]\) = expected system makespan.

### 3.2.3 Correlations among Resource Ready Times

Since \(G, T\) and \(P\) in equations (3.2) - (3.4) are assumed to be mutually independent and normally, (but not necessary identically), distributed random variables, \(R, A\) and \(D\) will also be random variables with correlations caused by the recursive relationships given in equations (3.1) - (3.4) (i.e., interactions among robot activities). The distribution of the maximum of a set of dependent random variables cannot, in general, be defined exactly, so the distribution of \(B\), which is the greatest of \(R, A\) and \(D\), in equation (3.1) can only be evaluated approximately. Following prior recursive modeling procedures, the joint distribution of \(B, R, A\) and \(D\) in equation (3.1) is approximated by the multivariate normal. Hence, Clark's (1961) equations may be used to approximate the moments of the beginning time of each robot activity.
3.2.3.1 Clark's Procedure

For two normally distributed random variables $X_1$ and $X_2$, with parameters $(\mu_1, \sigma_1)$ and $(\mu_2, \sigma_2)$, respectively, and correlation $\rho = r[X_1, X_2]$ ($\rho \neq 1$), Clark (1961) showed that the mean and variance of the random variable $Z = \max[X_1, X_2]$ are:

$$E[Z] = \mu_1 \Phi(\alpha) + \mu_2 \Phi(-\alpha) + a \phi(\alpha)$$

$$V[Z] = (\mu_1^2 + \sigma_1^2)\Phi(\alpha) + (\mu_2^2 + \sigma_2^2)\Phi(-\alpha) + (\mu_1 + \mu_2) a \phi(\alpha) - E^2[Z]$$

in which,

$$a^2 = \sigma_1^2 + \sigma_2^2 - 2 \rho \sigma_1 \sigma_2$$

$$\alpha = (\mu_1 - \mu_2) / a$$

$$\phi(\alpha) = \text{standard normal pdf evaluated at } \alpha$$

$$\Phi(\alpha) = \text{standard normal cdf evaluated at } \alpha.$$  

For the maximum of trivariate normal random variables, $\max[X_1, X_2, X_3]$, Clark (1961) suggests to express $\max[X_1, X_2, X_3]$ as $\max[\max(X_1, X_2), X_3] = \max[Z, X_3]$. By approximating the distribution of $Z = \max(X_1, X_2)$ as a normal, the equations given above could be used again to evaluate the moments of the maximum of $Z$ and $X_3$. Clark provides an equation to estimate the correlation between $Z$ and $X_3$:

$$r[\max(X_1, X_2), X_3] = [\sigma_1 \rho_1 \Phi(\alpha) + \sigma_2 \rho_2 \Phi(-\alpha)] / V^{1/2}[Z]$$

(3.5)
in which $\rho_1 = r[X_1, X_3]$ and $\rho_2 = r[X_2, X_3]$.

### 3.2.3.2 Notation

The following notation stores the relative information for each robot activity that is used in the following sections.

- $n = \text{individual robot activity index}; n=1, 2, ..., N.$
- $m = \text{system activity index}; m=1, 2, ..., M.$
- $R'_{nr}(cdj)$ denotes, for the $n$th activity of robot $r$, the lot $j$ that was moved from location $c$ to destination $d$.
- $J'_{j(nr)}$ denotes, for lot $j$, the $n$th activity of robot $r$ which last serviced it.
- $I'_{1(nr)}$ denotes, for location $1 \{1\mid 1=1,2,...,L\}$, the $n$th activity of robot $r$ which last serviced it (note that location $1$ is denoted as $c$ or $d$, depending upon its role in an activity).
- $X'_{m(nr)}$ gives, for system activity number $m$, the robot $r$ and its activity number, $n$.
- $X''_{nr}(m)$ gives, for the $n$th activity of robot $r$, the corresponding system activity number, $m$.
- $Y'_{r_q, r_m}(n'_{q})$ gives the last activity for robot $r_q, n'_{q}$, that robots $r_q$ and $r_m$ exchanged material.
- $Y''_{r_q, r_m}(n'_{m})$ gives the last activity for robot $r_m, n'_{m}$, that robots $r_q$ and $r_m$ exchanged material.
3.2.3.3 Computation of Correlations

In order to apply Clark's equations to estimate the mean and variance of activity beginning time, $B_{cdjr}$, in equation (3.1), the three pairwise correlations $\rho = r[R_{nr}, A_{cj}]$, $\rho_1 = r[R_{nr}, D_{dj}]$ and $\rho_2 = r[A_{cj}, D_{dj}]$ must be estimated. The time at which robot $r$ is ready for its $n$th activity, $R_{nr}$, is determined by the completion time of its last activity, $R_{n-1,r}(c^*d^*j^*)$, which is described by equation (3.3) (with $n$ replaced by $n-1$). Lot $j$ becomes ready at location $c$, $A_{cj}$, upon completion of an operation, which is initiated by robot activity $J'_j(n'r')$, according to equation (3.4). As described by equation (3.2), location $d$ is ready to receive another lot, $D_{dj}$, once the processed lot was unloaded by a robot activity, $I'_d(n''r'')$. Hence, the beginning time of robot $r$'s $n$th activity, $B_{cdjr}$, may be represented as:

$$B_{cdjr} = \max\{B_{c'd'^*j'^*n-1,r} + G_{c'd'^*n-1,r} + T_{c'd'^*n-1,r},$$
$$B_{c'jn'r'} + G_{c'jn'r'} + T_{c'jn'r'} + P_{cj},$$
$$B_{dd''j''n''r''} + G_{d'd''n''r''}\}$$

With the following definitions of "beta factors" (subscripts indicate a relationship to equation 3.2, 3.3 or 3.4), each of which is the ratio of the standard deviation of activity beginning time to that of the associated $D$, $R$ or $A$ ready time:

$$\beta_2(n'',r'') = V^{1/2}[B_{dd''j''n''r''}] / V^{1/2}[D_{dj}]$$
$$\beta_3(n-1,r) = V^{1/2}[B_{c'd'^*n-1,r}] / V^{1/2}[R_{nr}]$$
$$\beta_4(n',r') = V^{1/2}[B_{c'jn'r'}] / V^{1/2}[A_{cj}]$$
The required, pairwise correlations \( \rho, \rho_1 \) and \( \rho_2 \) may be calculated as:

\[
\rho = r[R,A] = r[R_{nr}, A_{cj}]
\]

\[
= r[B_{c}^{*}d_{j}^{*}, n-1,r + G_{c}^{*}j_{n-1,r} + T_{c}^{*}d_{j}^{*}, n-1,r + B_{c}^{*}c_{jn}^{*}r + G_{c}^{*}c_{jn}^{*}r + T_{c}^{*}c_{jn}^{*}r + P_{c}^{*}c_{j}^{*}r]
\]

\[
= r[B_{c}^{*}d_{j}^{*}, n-1,r, B_{c}^{*}c_{jn}^{*}r] \beta_3(n-1,r) \beta_4(n',r')
\] (3.6)

\[
\rho_1 = r[R,D] = r[R_{nr}, D_{dj}]
\]

\[
= r[B_{c}^{*}d_{j}^{*}, n-1,r + G_{c}^{*}j_{n-1,r} + T_{c}^{*}d_{j}^{*}, n-1,r, B_{d}^{*}d_{j}^{*}, n''r'' + G_{d}^{*}d_{j}^{*}, n''r'']
\]

\[
= r[B_{c}^{*}d_{j}^{*}, n-1,r, B_{d}^{*}d_{j}^{*}, n''r''] \beta_3(n-1,r) \beta_2(n'',r'')
\] (3.7)

\[
\rho_2 = r[A,D] = r[A_{cj}, D_{dj}]
\]

\[
= r[B_{c}^{*}c_{jn}^{*}r + G_{c}^{*}c_{jn}^{*}r + T_{c}^{*}c_{jn}^{*}r + P_{c}^{*}c_{j}^{*}r, B_{d}^{*}d_{j}^{*}, n''r'' + G_{d}^{*}d_{j}^{*}, n''r'']
\]

\[
= r[B_{c}^{*}c_{jn}^{*}r, B_{d}^{*}d_{j}^{*}, n''r''] \beta_4(n',r') \beta_2(n'',r'')
\] (3.8)

The correlations required by equation (3.5) may be estimated directly using equations (3.6) - (3.8), which may be obtained from correlations between certain activity beginning times and the associated beta factors. In some special cases, correlations \( \rho, \rho_1, \) and \( \rho_2 \) may be calculated even more easily. For example, if robot and location ready times are determined by the same robot activity beginning time (i.e., if \( r=r' \) and \( n-1=n' \))

\[
\rho = \sqrt{\frac{1}{2}[R_{nr}] / \sqrt{\frac{1}{2}[A_{cj}]}.
\]

In summary, to estimate the mean and variance of activity beginning time, \( B_{cdjnr} \), in equation (3.1), the three pairwise correlations \( \rho=r[R,A], \rho_1=r[R,D] \) and \( \rho_2=r[A,D] \) must be estimated. To estimate the pairwise correlations among resource ready times (i.e., R, A and D), the
pairwise correlations among the beginning times of the activities which initiate R, A and D must be determined. Upon evaluating equation (3.1) if any required correlation among the beginning times of activities was not calculated (and stored) previously, it must be estimated first. An efficient, "fill-in on the fly" procedure, which may calculate the unknown correlation between the beginning times of two activities, is described in the next subsection.

3.2.3.4 Fill-in on the Fly Procedure

The fill-in procedure, which works backwards over the completed sequence of activities, utilizes an efficient bookkeeping method to indicate which correlations are needed and how they are computed in an imbedded computational procedure. To describe this procedure, variables B, R, A and D are used as before, but subscripts are denoted relative to the fill-in procedure. Figure 3.1 (only a portion of the network is shown for clarity) illustrates the relationships among correlations that are used to compute the correlation between the beginning times of two activities, say \( r[B_q,B_1] \), which is required in evaluating equation (3.6), (3.7) or (3.8).

The correlation relationships depicted in Figure 3.1 may be described using the following notation:

\[
\begin{align*}
  m & = \text{index for nodes in the network; } m=1, 2, \ldots, M. \\
  P[m] & = \text{index for the parent node of node } m. \\
  N[m,k] & = \text{index for the } k^{th} \text{ son of node } m; \ k=1, 2, 3.
\end{align*}
\]
Figure 3.1 Correlation Relationships.
$Q[p,k|m]$ denotes, for given node $m$, its parent node $p=P[m]$ and the $k^{th}$ son of $p$ so that $N[p,k]=m$.

Relationships among these notations are depicted in Figure 3.2. In Figure 3.1, a node is used to represent the correlation between the beginning times of two activities. Each node $m$, has three sons: $(m,1)$, $(m,2)$, $(m,3)$ [obtained from $N(m,k)$, $k=1,2,3$] which correspond to the $R$, $A$ and $D$ times (respectively) that define, through equation (3.1), the beginning time of the activity represented by node $m$. For example, the parent of node $m=5$ in Figure 3.1 is $P[5]=2$, its sons are designated $(5,1)$, $(5,2)$ and $(5,3)$; and they are numbered $N[5,1]=14$, $N[5,2]=15$ and $N[5,3]=16$. The function $Q[p,k|m]$, for $m=5$, also indicates that $(p,k) = (2,1)$ since node 5 is the first son of node 2. $N(m,k)$ is defined as 0 if the ready time which corresponds to node $N(m,k)$ is zero.

The correlation relationships in the fill-in on the fly procedure may now be described using the node relationships in the network just discussed. The correlation $r[B_q,B_m]$ ($q$ represents any system activity number with $q \leq m$) may be calculated using equations (3.1) - (3.4):

$$r[B_q,B_m] = r[B_q, \max(R_m, A_m, D_m)]$$

$$= r[B_q, \max(B_{m1} + G_{m1} + T_{m1}, \max(B_{m2} + G_{m2} + T_{m2} + P_{m2}, B_{m3} + G_{m3}))]$$

in which $B_{mk}$ ($k=1,2,3$) represents the activities that initiate the ready times of the robot ($R_m$), the lot ($A_m$) and the destination ($D_m$) (respectively) which determine the beginning time of the activity, $B_m$, represented by
\[ p = P[m] \]

\[ m = N[p, k] \]

\( (p, k) \) from \( Q[pk|m] \)

\[ P[N(m,k)] = m \text{ for } k = 1, 2, 3 \]

Figure 3.2 Notation for Network.
node \( m \). If pairwise correlations \( r[B_q,B_m1], r[B_q,B_m2] \), and \( r[B_q,B_m3] \) are known, \( r[B_q,B_m] \) may be calculated from equation (3.5) with \( \rho, \rho_1 \), and \( \rho_2 \) defined as:

\[
\rho = r[B_{m1}+G_{m1}+T_{m1}, \max(B_{m2}+G_{m2}+T_{m2}+P_{m2}, B_{m3}+G_{m3})] \quad (3.10)
\]

\[
\rho_1 = r[B_q,B_{m1}] \beta_3(m,1) \quad (3.11)
\]

\[
\rho_2 = r[B_q, \max(B_{m2} + G_{m2} + T_{m2} + P_{m2}, B_{m3} + G_{m3})] \quad (3.12)
\]

\( \rho \) was calculated in determining \( B_m \) and \( \rho_2 \) may be found using equation (3.5) with

\[
\rho' = r[B_{m2}, B_{m3}] \beta_4(m,2) \beta_2(m,3) \quad (3.13)
\]

\[
\rho'_1 = r[B_q, B_{m2}] \beta_4(m,2) \quad (3.14)
\]

\[
\rho'_2 = r[B_q, B_{m3}] \beta_2(m,3) \quad (3.15)
\]

If any resource ready time, \( R_m, A_m \), or \( D_m \) in equation (3.9) is zero, the correlation \( r[B_q,B_m] \) may be obtained directly from equations (3.13) - (3.15).

This fill-in procedure is a depth-first procedure and, therefore, always expands the larger activity number first. For example, if \( r[B_q,B_{1,1}] \) is not known and the activity number of (1,1) is larger than \( q \), it may be found by first expanding \( B_{1,1} \) to obtain \( B_{2,k} \) (\( k=1,2,3 \)), then evaluating \( r[B_q,B_{2,k}] \) (\( k=1,2,3 \)). If all correlations \( r[B_q,B_{2,k}] \) (\( k=1,2,3 \)) are known, equation (3.9) may be used to calculate \( r[B_q,B_{1,1}] \). If, for example, \( r[B_q,B_{2,1}] \) and \( r[B_q,B_{2,2}] \) are not known, we will first evaluate \( r[B_q,B_{2,1}] \), then consider \( r[B_q,B_{2,2}] \) (this is the idea of depth-first). The (backward) search for known correlations continues until all those required correlations are found from
prior calculations or until activities are met which are very "remote" relative to $B_q$ and the pairwise correlation can be assumed to be zero. The remoteness between two (system) activities, $q$ and $m$, may be defined by either of the following two procedures:

1. Procedure SIMPLE
   (a) Set $L =$ the difference between two activity numbers $(m-q)$

2. Procedure REMOTE
   (a) Determine the robot and robot activity numbers corresponding to system activity number $q$, $X'_q(n_qr_q)$, and with system activity number $m$, $X'_m(n_mr_m)$.

   (b) If $r_q = r_m$, set $L = (n_m - n_q)$ and Stop. (both activities are performed by the same robot).

   (c) If $r_q \neq r_m$, determine the most recent activities of the two robots in which they exchange material directly. Denote this activity, $Y'_{r_q, r_m}(n'_q)$ (for robot $r_q$, $Y''_{r_q, r_m}(n'_m)$) for robot $r_m$ as $n_q$ ($n'_m$).
   Set $L = (n_q - n'_q) + (n'_m - n_m)$ and Stop.

In order to raise efficiency of the fill-in on the fly procedure for estimating any unknown correlation, say $r[B_q, B_1]$, algorithm SEARCH was based on the concept of correlation relationships in the network discussed above. Either procedure SIMPLE or REMOTE may be used in Step 2 to compare $L$ with $L_{max}$, a predetermined criterion of remoteness.

1. If the correlation for node 1, $r[B_q, B_1]$, is known, go to Step 8 (correlation need not be calculated)
Else, set $m = 1$ and go to Step 3.

(2) Use procedure SIMPLE or REMOTE to determine $L$.
   If $L > L_{\text{max}}$, set $m = \text{P}[m]$ and $r[B_q, B_{mk}] = 0$, then go to Step 5
   (eliminate remote nodes).

(3) Define sons of $m$, $\{N[m,k] : k=1,2,3\}$ and family relationships,
   $\{P[N(m,k)] : k=1,2,3\}$ and $\{Q[p,k \mid N(m,k)] : k=1,2,3\}$.
   Set $k = 1$ (the firstborn son of node $m$).

(4) If the correlation $r[B_q,B_{mk}]$ is unknown (for son $mk$), set
    temporary index $m'' = \max\{Q[p,k \mid N(m,k)], q\}$, then set
    $q = \min\{Q[p,k \mid N(m,k)], q\}$ and $Q[p,k \mid N(m,k)] = m''$ and $m = N[m,k]$ (so that activity $m''$ is expanded) and go to Step 2 (to
    calculate the required correlation for this $m$).

(5) Set $k = k+1$
   If $N[m,k] > 0$ and $k \leq 3$, go to Step 4 (check the next-born son
   of $m$)
   If $k < 3$, go to Step 5.

(6) Use $Q[pk \mid m]$ to determine the $pk$ relationship for $m$.
    Calculate $r[B_q,B_{pk}]$ using equation (3.9); store it (correlations
    for all sons of $m$ are now known).

(7) If $p > q$, set $m = p$ and go to Step 5.
    (to check brothers of $pk$; $k$ is known from Step 6).

(8) STOP (the required correlation, $r[B_q,B_1]$ is known).

Upon applying algorithm SEARCH to evaluate correlation of node
$m$, family relationships of node $m$ must be constructed first (step 3), then
the correlation of the first born son, $r[B_q,B_{mk}]$ ($k=1$), of node $m$ is
considered. If $r[B_q,B_{mk}]$ is unknown and cannot be set to zero from step 2,
node N(m,k) must be expanded to evaluate correlation $r[B_q,B_{mk}]$ first (step 4); this search procedure continues until $r[B_q,B_{mk}]$ is obtained. Then, the correlation of the next born son, $r[B_q,B_{mk}]$ ($k=k+1$), of node m may be evaluated (step 5) following the procedure described above. After all three correlations, $r[B_q,B_{mk}]$ ($k=1,2,3$), required by node m are obtained, using function $Q[p,k|m]$ to determine the pk relationships for node m and applying equation (3.9) to calculate correlation of node m, $r[B_q,B_{pk}]$, (step 6). If p is equal to the first node q, correlation $r[B_q,B_1]$ is obtained, otherwise, continuing this search procedure for node m's brothers (i.e., set $m=p$ and go to step 5). This depth-first procedure (depth is achieved at step 4, breadth at step 5) can efficiently estimate the required correlation because it only evaluates the necessary correlations.

3.3 Logic of implementing the Recursion Procedure

The recursion model described in this dissertation is an event-based model which parallels certain concepts of discrete-event simulation, and ranks events according to the expected time at which they will occur. Ordering events according to the expected time of occurrence describes, in some sense, the central tendency of the process. A corresponding discrete-event simulation model, which was used to evaluate the accuracy of recursion model, ranks events according to "realized" time (a random variate). By replicating, simulation may generate a number of the possible sequences of events (or activities); the recursion model describes only one sequence, but any sequence could be modeled.
The recursion and simulation models also use the same filing system to initiate robot activities; there are three different types of files:

1. **Destination file**: this file is used to store information (e.g., current location) of each lot which is seeking to be serviced by a station; each station has two types of destination files: one, IB, stores lots in the station's input buffer, and the other, OB, stores lots currently located at "other place" (e.g., output buffers of other stations).

2. **Solicitation file**: each robot has a solicitation file to organize calls for service according to priority: output buffers, idle machines and input buffers. Within each priority class, calls are ranked according to the (expected) first-come-first-served (FCFS) rule. An output buffer (of a station) call is filed only when a lot is waiting to be unloaded from one of the machines in the station and this buffer is ready to receive it.

3. **Event files**: there are three types of events; departure, arrival and completion events. Basically, resource ready times are "realized" by these events. For example, a destination is ready for the next operation when a lot was unloaded from it (i.e., the departure event is realized) and a robot is ready for the next movement after it loads a lot at a location (i.e., the arrival event is realized). Finally, a lot is ready for its next operation after completing service at its current location (i.e., a completion event occurs).
Based on the filing system described above, the logic of implementing the recursion procedure is summarized in the following four steps:

(1) Initialization:
   a. Initiate solicitation file for each robot so that all the available destinations which need to be serviced by this robot are filed.
   b. Initiate destination file for each station so that all the available lots which are seeking this destination are filed.

(2) Determination of the beginning time of a robot activity:
   a. select robot r which is idle or has just completed its (n-1)th activity (i.e., robot r becomes ready for the nth movement).
   b. check robot r's solicitation file and select the priority destination d (e.g., I/O buffer or a machine at a station). If the input buffer or a machine at a station is to be serviced, check the destination file of this station and select the priority lot j (currently located at c).
   c. apply equation (3.1) to describe the beginning time of the next activity, B_{cdjr}, according to the resource ready times obtained in (2b).

(3) Determination of resource ready times:
   a. apply equations (3.2) - (3.4) to schedule the (expected) realization time for (i) the departure event which readies location c for the next lot, D_{cj+}, (ii) the arrival event which readies robot r for its next activity, R_{n+1,r}, and (iii) the completion event which readies lot j for its next operation, A_{dj}.
b. file these events into the EVENT file.

(4) If the EVENT file is empty, then STOP (all lots were completed)
Else, select the priority event from the EVENT file; if it is
a. a departure event, then file location c into the corresponding solicitation file and go to (2).
b. an arrival event, then go to (2).
c. a completion event, then file lot j into the destination file associated with the next station on the lot's routing and go to (2).

Both recursion and simulation models use this logic to schedule events and estimate system performance (e.g., makespan). The simulation model may generate a number of the possible sequences of activities, but the recursion model gives only one sequence. In order to compare both models on the same basis, all simulation replications should use the same, "fixed", sequence of activities generated by the recursion model so that differences are due to model accuracy, not to differences between sequences.

Because of the filing system and the logic of implementing the recursion model, a broad variety of features in a cellular assembly system could be modeled:

a. Job-shop routing: once a lot is completed at its current location, it is filed into the destination file associated with the next station on this lot's routing. Hence, any type of lot routings may be modeled.
b. Finite input/output buffers, parallel machines and material handling: the recursion model keeps track of relationships among robot activities, each activity starts and ends from buffers and/or machines. Hence, these features could be modeled.

c. Lot/machine sequencing: destination (solicitation) file ranks lots (locations) according to the predetermined lot (machine) sequencing rule.

3.4 **Summary of Model 1**

In Model 1, the recursion procedure is extended to model material flow in cellular, small-lot assembly systems with a broad variety of features, including job-shop routing, finite input/output buffers, lot/machine sequencing, material handling and parallel machines. The recursion model parallels certain concepts of discrete-event simulation, and ranks events according to the **expected time** at which they will occur. An efficient computational procedure was developed to evaluate correlations among resource ready times, so that activity (e.g., robot operation) beginning times may be evaluated efficiently.
CHAPTER IV

NUMERICAL EVALUATION OF MODEL 1

4.1 Introduction

The recursion procedure described in Chapter 3 is evaluated in this chapter. In Section 4.2, model accuracy is evaluated relative to a number of system features (e.g., lot routing, lot sequencing, parallel machines and finite buffer capacities). In Section 4.3, the recursion model is then demonstrated in application to an hypothetical industrial setting.

4.2 Model Evaluation Relative to Different System Features

Figure 4.1 depicts the hypothetical robotic cell, which consists of six stations and three robots, used to evaluate the accuracy and computational characteristics of the recursion model. Detailed data which describes the robotic cell is given in Table 4.1, including the number of machines at each station, robot service times, part families, lot routings and mean processing times for each part family under each flow structure.

4.2.1 Different Flow Structures

Earlier recursion models were relatively less accurate when mean processing times were "balanced" [Wilhelm and Ahmadi-Marandi (1982),
Figure 4.1 Hypothetical Robotic Cell.
Table 4.1 Configuration for Evaluating Recursion Model Accuracy.

**SYSTEM:**
1 cell, 3 robots, and 6 stations (see Figure 3)

**STATIONS:**

<table>
<thead>
<tr>
<th>Item</th>
<th>Station Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5  6</td>
</tr>
<tr>
<td>Number of Machines</td>
<td>1  1  1  2  3  4</td>
</tr>
<tr>
<td>Input Buffer Capacity</td>
<td>4  4  4  4  4  4</td>
</tr>
<tr>
<td>Output Buffer Capacity</td>
<td>1  1  1  1  1  1</td>
</tr>
</tbody>
</table>

**ROBOTS:**
- $E[T_2] = \text{mean travel/load and travel/unload time within cell (between different stations)} = 21 \text{ sec}$
- $E[T_1] = E[T_3] = \text{mean travel/load and travel/unload time between cell and system buffers} = 36 \text{ sec}$
- $E[G] = \text{mean travel/load and travel/unload time inside one station} = 21 \text{ sec}$
- $CV = \text{coefficient of variation for all robot travel/grasp/release times} = 0.25$

**JOBS:**
- $J = 100$ lots are processed; job family is randomly generated
- $CV = \text{coefficient of variation of all processing times is } 0.25$

**ROUTINGS FOR JOB FAMILIES AND MEAN PROCESSING TIMES:**

<table>
<thead>
<tr>
<th>Job Family</th>
<th>Job Station Routing</th>
<th>E[$P_i$] in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Routing Position</td>
<td>1  2  3  4  5  6</td>
</tr>
<tr>
<td></td>
<td>Routing Position</td>
<td>1  2  3  4  5  6</td>
</tr>
<tr>
<td></td>
<td>FL - Flowline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1  1  2  3  4  5  6</td>
<td>10  10  10  10  10  10</td>
</tr>
<tr>
<td></td>
<td>GFL - Generalized Flowline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1  1  3  5</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>2  1  4  5</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>3  2  4  6</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>4  2  3  6</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>JS - Job Shop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1  1  3  5</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>2  4  2  6</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>3  2  4  6</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>4  3  1  5</td>
<td>20  20  20</td>
</tr>
<tr>
<td></td>
<td>PM - Parallel Machine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1  1  2  3  4  5  6</td>
<td>5  5  5  5  10  15  20</td>
</tr>
</tbody>
</table>

PM - Number of Machines in Parallel at each Station

| 1  1  1  2  3  4 |
Saboo and Wilhelm (1986)], and when a large number of machines are used at individual stations [Saboo, Wang and Wilhelm (1986)]. Balanced mean processing times for lots at all stations and parallel machines (PM) cases are used to evaluate the accuracy of this new recursion procedure. Besides, test cases were also defined to consider three workflow structures; the flowline (FL), the generalized flowline (GF) and the job shop (JS). Both FL1 and FL2 use the FL routing structure; the former has I/O buffer capacities of zero at each station while the latter incorporates the capacities indicated in Table 4.1.

Since the recursion model is an approximation model, errors of approximation are expected to increase as the number of activities increases. Estimates of the expected value of system makespan are, therefore, used to evaluate the accuracy of the recursion model. Recursion estimates are compared with 95% confidence intervals derived from the simulation model. Since the width of a confidence interval depends upon the number of replications, it gives only a relative measure of performance. In order to compare recursion estimates to relatively precise simulation estimates, the simulation model is replicated until the half width of the 95% confidence interval is less than 0.5% of the estimated mean value of makespan. Other means for evaluating the accuracy of the recursion model are the average (MD%) and mean square (MSD%) percent differences between recursion and simulation estimates of the mean values of all activity beginning times over an entire schedule of J lots.

The effect of Lmax, a predetermined criterion of remoteness between two activities, on the estimates of expected makespan for each of
the five flow structures (i.e., FL1, FL2, GFL, JS and PM) is listed in Table 4.2, which shows that a smaller (larger) value of Lmax would require less (more) runtime, but may give a poorer (better) estimate of system performance. From these preliminary results, an Lmax value of 60 is used in subsequent tests to give a reasonable balance between runtime and accuracy.

Table 4.3 gives an analysis of model performance for each of the five flow structures. In practical assembly systems, many schedulers apply deterministic methods to schedule assembly operations. In order to study the impact of the variance of operation time, the estimate of makespan for each of the five flow structures was calculated by assuming that the variances of operation times for all lots at all stations are zero. Results in Table 4.3 indicate that using this deterministic method (i.e., neglecting the variance of operation time) gives a very short system makespan which is impossible to be achieved.

Saboo and Wilhelm (1986) indicate that setting the pairwise correlation between resource ready times to 1.0 (0.0), equation (3.1) gives the smallest (largest) possible value for the expected beginning time of an operation. These limiting values are also listed in Table 4.3 to highlight the importance of computing accurate estimates of correlations for each operation (or activity).

As mentioned earlier, the Expected-First-Come-First-Serve (EFCFS) rule is used by the recursion model to generate the sequence of activities in the schedule, and this "fixed" sequence is also used by all simulation replications. Because all replications utilize the same sequence, the
Table 4.2 The Effect of Lmax for Different Job Routings.

<table>
<thead>
<tr>
<th>Job routing</th>
<th>Lmax</th>
<th>( \bar{M} )</th>
<th>Run Time</th>
<th>**</th>
<th>Simulation+ Fixed sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \bar{M} )</td>
<td></td>
<td></td>
<td>( \bar{M} )</td>
</tr>
<tr>
<td>F.L.1</td>
<td>20</td>
<td>1454.20</td>
<td>7.186</td>
<td></td>
<td>1450.71 54.515</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1454.20</td>
<td>7.245</td>
<td></td>
<td>1865.69 53.673</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1454.20</td>
<td>7.263</td>
<td></td>
<td>1865.69 53.673</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1454.20</td>
<td>12.955</td>
<td></td>
<td>1865.69 53.673</td>
</tr>
<tr>
<td>F.L.2</td>
<td>20</td>
<td>1865.69</td>
<td>19.366</td>
<td></td>
<td>1869.92 53.673</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1865.69</td>
<td>19.561</td>
<td></td>
<td>1869.92 53.673</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1865.69</td>
<td>22.100</td>
<td></td>
<td>1869.92 53.673</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1865.69</td>
<td>22.327</td>
<td></td>
<td>1869.92 53.673</td>
</tr>
<tr>
<td>G.F.L</td>
<td>20</td>
<td>1728.60</td>
<td>12.895</td>
<td></td>
<td>1690.92 81.123</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1701.17</td>
<td>14.483</td>
<td></td>
<td>1690.92 81.123</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1701.06</td>
<td>16.773</td>
<td></td>
<td>1690.92 81.123</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1701.07</td>
<td>19.332</td>
<td></td>
<td>1690.92 81.123</td>
</tr>
<tr>
<td>J.S.</td>
<td>20</td>
<td>1717.04</td>
<td>12.971</td>
<td></td>
<td>1667.42 92.618</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1677.50</td>
<td>15.845</td>
<td></td>
<td>1667.42 92.618</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1677.56</td>
<td>19.658</td>
<td></td>
<td>1667.42 92.618</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1677.57</td>
<td>22.456</td>
<td></td>
<td>1667.42 92.618</td>
</tr>
<tr>
<td>P.M.</td>
<td>20</td>
<td>1670.91</td>
<td>17.951</td>
<td></td>
<td>1615.86 74.409</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1592.44</td>
<td>25.619</td>
<td></td>
<td>1615.86 74.409</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1619.01</td>
<td>27.486</td>
<td></td>
<td>1615.86 74.409</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1619.16</td>
<td>27.613</td>
<td></td>
<td>1615.86 74.409</td>
</tr>
</tbody>
</table>

\( \bar{M} \) average system makespan

+ use antithetic variance reduction technique

** CPU seconds on IBM 3081-D
### Table 4.3 Comparison of Estimates for Different Job Routings.

<table>
<thead>
<tr>
<th>Job Routing</th>
<th>RECURSION</th>
<th>SIMULATION+</th>
<th>ROBOT OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deterministic</td>
<td>Fixed Sequence</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\rho = 1.0$</td>
<td>$\rho = 0.0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run Time</td>
<td>U</td>
<td>AHW%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F.L.1</td>
<td>1195.35</td>
<td>1195.35</td>
<td>1651.07</td>
</tr>
<tr>
<td>F.L.2</td>
<td>1543.07</td>
<td>1543.08</td>
<td>1960.81</td>
</tr>
<tr>
<td>G.F.L</td>
<td>1360.68</td>
<td>1359.41</td>
<td>1858.97</td>
</tr>
<tr>
<td>J.S.</td>
<td>1358.33</td>
<td>1313.29</td>
<td>1841.66</td>
</tr>
<tr>
<td>P.M.</td>
<td>1107.19</td>
<td>1102.65</td>
<td>1989.62</td>
</tr>
</tbody>
</table>

- $M$ = Average system makespan
- $S$ = Standard deviation of system makespan
- $L$ = Lower limit of 95% confidence interval
- $U$ = Upper limit of 95% confidence interval
- $\text{AHW\%} = \text{Average half-width of the confidence interval (expressed as percent)}$
- $\Delta\% = 100 \left( \frac{M_{\text{simulation}} - M_{\text{recursion}}}{M_{\text{simulation}}} \right)$
- $\text{MD\%} = \text{Mean difference of robot operation beginning time between simulation and recursion model}$
- $\text{MSD\%} = \text{Mean square difference of robot operation beginning time between simulation and recursion model}$
- $+$ = Use antithetic variance reduction technique
- $*$ = Recursion estimate, $M$, falls outside 95% confidence interval of fixed-sequence simulation
- $**$ = CPU seconds on IBM 3081-D
antithetic variate (AV) variance reduction technique [Fishman (1978)] may be applied to reduce the number of replications required to achieve the prespecified level of precision.

Three of the five recursion estimates of the expected makespan fall within the 95% confidence intervals; all MD% and MSD% measures are relatively small. For both balanced mean processing times and parallel machines (PM) cases for which prior recursion models did not produce good estimates, this new algorithm gives accurate estimates. The relative difference between recursion and simulation estimates of expected activity beginning time, depicted in Figure 4.2, converges to a small limiting value as the number of activities increases. Because of the use of the AV method, relatively few simulation replications were required. Recursion model runtime is approximately that of 7.6 simulation replications in these five cases.

4.2.2 Finite Buffer Capacities

Studying buffer capacities was another means of evaluating the recursion model accuracy and applicability as a decision aid. The FL routing structure was used in this analysis. Table 4.4 specifies mean robot service times and the mean processing times for each part family at each station, along with a coefficient of variation (cv) of 0.25 for all operations. An Lmax of 40 was selected in this analysis.

The effect of input/output buffers on the system performance (e.g., makespan) may be noted in Table 4.5, which indicates that buffers reduce
Figure 4.2 Relative Difference Between Recursion and Simulation Estimates of Expected Activity Beginning Time vs Activity Number (Hypothetical Robotic Cell).
Table 4.4 System Design for I/O Buffer Analysis.

ROBOTS:

\[ E[T_2] = \text{mean travel/load and travel/unload time within cell (between different stations)} = 21 \text{ sec} \]
\[ E[T_1] = E[T_3] = \text{mean travel/load and travel/unload time between cell and system buffers} = 36 \text{ sec} \]
\[ E[G] = \text{mean travel/load and travel/unload time inside one station} = 3 \text{ sec} \]

ROUTINGS FOR JOB FAMILIES AND MEAN PROCESSING TIMES:

<table>
<thead>
<tr>
<th>Job</th>
<th>Station Routings</th>
<th>Family Routing Position</th>
<th>( E[P_i] \text{ in min.} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 2 3 4 5 6 1</td>
<td>2 3 4 5 6 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 2 3 4 5 6 1</td>
<td>2 3 4 5 6 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1 2 3 4 5 6 1</td>
<td>2 3 4 5 6 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 2 3 4 5 6 1</td>
<td>2 3 4 5 6 1</td>
<td></td>
</tr>
</tbody>
</table>

\( E[P_1] = 4 \text{ sec} \)
\( E[P_2] = 8 \text{ sec} \)
\( E[P_3] = 12 \text{ sec} \)
\( E[P_4] = 16 \text{ sec} \)
Table 4.5 The Effect of I/O Buffers on System Performance.

<table>
<thead>
<tr>
<th>CASE</th>
<th>IB</th>
<th>OB</th>
<th>Deterministic Estimate</th>
<th>P=1.0</th>
<th>P=0.0</th>
<th>Lmax=4n Run</th>
<th>**</th>
<th># of Replications</th>
<th>**</th>
<th>**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M M M S Time M L U AHW% Δ%</td>
<td>Repli-</td>
<td>Run</td>
<td>Run</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1657.95</td>
<td>1658.04</td>
<td>2002.83</td>
<td>1808.72</td>
<td>25.41</td>
<td>7.566</td>
<td>1801.69</td>
<td>1797.94</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>10</td>
<td>1263.01</td>
<td>1262.02</td>
<td>1762.27</td>
<td>1600.76</td>
<td>17.33</td>
<td>16.168</td>
<td>1588.09</td>
<td>1580.63</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0</td>
<td>1253.06</td>
<td>1252.84</td>
<td>1706.51</td>
<td>1468.81</td>
<td>16.27</td>
<td>15.021</td>
<td>1464.70</td>
<td>1459.70</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1199.49</td>
<td>1201.42</td>
<td>1710.46</td>
<td>1499.94</td>
<td>17.41</td>
<td>28.287</td>
<td>1478.97</td>
<td>1472.20</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1192.78</td>
<td>1190.37</td>
<td>1688.48</td>
<td>1500.04</td>
<td>14.08</td>
<td>28.10</td>
<td>1480.85</td>
<td>1473.81</td>
</tr>
</tbody>
</table>

M = Average system makespan
S = Standard deviation of system makespan
L = Lower Limit of 95% confidence interval
U = Upper limit of 95% confidence interval

AHW% = Average half-width of the confidence interval (expressed as percent)

Δ% = 100 (M simulation - M recursion)/(M simulation)

+ = Use antithetic variance reduction technique

• = Recursion estimate, M, falls outside 95% confidence interval of fixed-sequence simulation

** = CPU seconds on IBM 3081-D
makespan dramatically. The input buffers (case 3) have a more significant influence on makespan than the output buffers (case 2). Small increases in the buffer capacities (cases 1 and 4) may significantly reduce makespan, and adequate buffer capacities (case 5) give best system performance. To increase efficiency of system performance, buffer capacities should be determined carefully.

Although only one of the five recursion estimates of expected makespan falls within the 95% confidence intervals obtained from fixed-sequence simulation estimates, $\Delta\%$, the ratio of the difference of mean system makespan between simulation and recursion models to mean system makespan obtained from simulation model, seems to be acceptable for practical decision making purposes. Compared to the recursion and fixed-sequence simulation estimates, the non-fixed sequence simulation estimate of system makespan is much less, because it does not impose delays to assure a particular sequence of events (or activities). Results of the non-fixed sequence simulation can only represent an average system performance over all replications, not any particular sequence of activities as the recursion model does.

In order to study the effect of sequencing rules on system performance, a new rule, which ranks all calls in robot solicitation files according to the EFCFS rule and thereby eliminates priorities among machines and I/O buffers, was used to generate another sequence of activities for case 4 in Table 4.5. Results from the recursion model indicate that an appropriate sequencing rule gives better system performance. For example, this new sequence has more robot activities (2100) and less
expected system makespan (1491.13) compared with those obtained from the original sequencing rule.

4.3 Model Application in an Hypothetical Industrial Setting

The circuit card assembly (CCA) example described in Rockwell and Wilhelm (1988) was used to demonstrate application of the recursion model in an hypothetical industrial setting. The configuration of the hypothetical robotic cells is depicted in Figure 4.3, and described in Table 4.6, including part families processed at each station and estimates of resource utilization. Robot service times are given in Table 4.7. Detailed data which describes CCA family composition and mix is given in Table 4.8, including production output rate [which is 100 lots (CCA's) per day], the probability that a lot be assigned to a CCA family, the parameters of determining the total number of components on each circuit card (generated randomly from the uniform distribution), and the parameters for determining the number of components to be inserted at each station (generated randomly from a multinomial distribution). The expected setup time for each lot (CCA) and the expected insertion time per component at each station are specified in Table 4.9.

A factorial experiment was designed to test the accuracy of the recursion model relative to three factors (buffer capacity, number of machines at a station, and degree of balance in mean processing times) as specified in Table 4.10. Even though a large number of robot activities were performed, six of the eight recursion estimates of the expected
Figure 4.3 System Configuration of an Industrial Setting.
Table 4.6 System Design for Industrial Setting.

<table>
<thead>
<tr>
<th>STATIONS</th>
<th>#Mach</th>
<th>MHD?</th>
<th>Part Families</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>UT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CeU #1</td>
<td></td>
<td></td>
<td>MHD ----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100% max</td>
</tr>
<tr>
<td>1. DIP</td>
<td>2</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>40%</td>
</tr>
<tr>
<td>2. Axial</td>
<td>2</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>32%</td>
</tr>
<tr>
<td>3. Radial</td>
<td>1</td>
<td>No</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>29%</td>
</tr>
<tr>
<td>4. Multiple</td>
<td>1</td>
<td>No</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>44%</td>
</tr>
<tr>
<td>5. Pin</td>
<td>1</td>
<td>No</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18%</td>
</tr>
<tr>
<td>7. Robotic</td>
<td>2</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>62%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATIONS</th>
<th>#Mach</th>
<th>MHD?</th>
<th>Part Families</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>UT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CeU #2</td>
<td></td>
<td></td>
<td>MHD ----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18% max</td>
</tr>
<tr>
<td>1. Manual</td>
<td>2</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>45%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATIONS</th>
<th>#Mach</th>
<th>MHD?</th>
<th>Part Families</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>UT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CeU #3</td>
<td></td>
<td></td>
<td>MHD ----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75% max</td>
</tr>
<tr>
<td>1. Pre-solder</td>
<td>4</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>69%</td>
</tr>
<tr>
<td>2. Flow Solder</td>
<td>8</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>38%</td>
</tr>
</tbody>
</table>

Between Cell MHD----------| MHD----------| 72% max |

* UT = Utilizations which are based on a thruput of 30 CCA per hour

Table 4.7 Expected Robot Service Times.

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>GRASP (sec)</th>
<th>RELEASE (ft/sec)</th>
<th>TRAVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN CELLS</td>
<td>6.0</td>
<td>6.0</td>
<td>30.0</td>
</tr>
<tr>
<td>WITHIN CELLS</td>
<td>6.0</td>
<td>6.0</td>
<td>15.0</td>
</tr>
<tr>
<td>WITHIN STATION</td>
<td>6.0</td>
<td>6.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>
Table 4.8 CCA Family Composition and Mix.

<table>
<thead>
<tr>
<th>STATION</th>
<th>FAMILY 1</th>
<th>FAMILY 2</th>
<th>FAMILY 3</th>
<th>FAMILY 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIP</td>
<td>0.17</td>
<td>0.30</td>
<td>0.30</td>
<td>--</td>
</tr>
<tr>
<td>AXIAL</td>
<td>0.35</td>
<td>0.40</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>RADIAL</td>
<td>0.23</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>MULTIPLE</td>
<td>--</td>
<td>0.20</td>
<td>0.20</td>
<td>--</td>
</tr>
<tr>
<td>PIN</td>
<td>0.17</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ROBOTIC</td>
<td>0.08</td>
<td>--</td>
<td>0.50</td>
<td>--</td>
</tr>
<tr>
<td>MANUAL</td>
<td>--</td>
<td>0.10</td>
<td>--</td>
<td>1.00</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF COMPONENTS (DISCRETE UNIFORM DISTRIBUTION)

| LOWER BOUND | 50 | 100 | 125 | 20 |
| UPPER BOUND | 400 | 300 | 200 | 150 |
| MEAN        | 225 | 200 | 163 | 85 |

PRODUCTION VOLUME

| Fraction of 100/day | 0.4 | 0.3 | 0.2 | 0.1 |

Table 4.9 Expected Setup and Insertion Times.

<table>
<thead>
<tr>
<th>STATION</th>
<th>NUMBER OF MACHINES</th>
<th>TIME PER COMPONENT (sec)</th>
<th>TIME PER LOT (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIP</td>
<td>2</td>
<td>1.8</td>
<td>20</td>
</tr>
<tr>
<td>AXIAL</td>
<td>2</td>
<td>0.9</td>
<td>30</td>
</tr>
<tr>
<td>RADIAL</td>
<td>1</td>
<td>1.1</td>
<td>30</td>
</tr>
<tr>
<td>MULTIPLE</td>
<td>1</td>
<td>1.8</td>
<td>40</td>
</tr>
<tr>
<td>PIN</td>
<td>1</td>
<td>0.9</td>
<td>20</td>
</tr>
<tr>
<td>ROBOTIC</td>
<td>2</td>
<td>5.8</td>
<td>20</td>
</tr>
<tr>
<td>MANUAL</td>
<td>2</td>
<td>5.8</td>
<td>60</td>
</tr>
<tr>
<td>PRE-SOLDER</td>
<td>4</td>
<td>--</td>
<td>90</td>
</tr>
<tr>
<td>FLOW SOLDER</td>
<td>8</td>
<td>--</td>
<td>180</td>
</tr>
</tbody>
</table>
Table 4.10 Comparison of Estimates in an Industrial Setting.

<table>
<thead>
<tr>
<th>Case</th>
<th>Factor</th>
<th>Recursion</th>
<th>Simulation</th>
<th>Robot Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Levels</td>
<td>p=1.0</td>
<td>p=0.0</td>
<td>L_{max}=60</td>
</tr>
<tr>
<td>B M D</td>
<td>Estimates</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>1 - - -</td>
<td>418.47</td>
<td>418.50</td>
<td>459.78</td>
<td>429.07</td>
</tr>
<tr>
<td>2 + - -</td>
<td>481.97</td>
<td>481.97</td>
<td>532.37</td>
<td>482.48</td>
</tr>
<tr>
<td>3 - + -</td>
<td>404.57</td>
<td>403.50</td>
<td>443.86</td>
<td>416.49</td>
</tr>
<tr>
<td>4 + + -</td>
<td>335.92</td>
<td>335.96</td>
<td>346.05</td>
<td>338.57</td>
</tr>
<tr>
<td>5 - - +</td>
<td>410.94</td>
<td>418.71</td>
<td>492.57</td>
<td>458.18</td>
</tr>
<tr>
<td>6 + + +</td>
<td>474.77</td>
<td>475.09</td>
<td>533.71</td>
<td>481.80</td>
</tr>
<tr>
<td>7 - + +</td>
<td>388.78</td>
<td>393.44</td>
<td>452.91</td>
<td>412.86</td>
</tr>
<tr>
<td>8 + + +</td>
<td>317.78</td>
<td>317.91</td>
<td>350.57</td>
<td>345.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factor</th>
<th>-</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = Buffer capacity</td>
<td>IB=4</td>
<td>OB=1 for each machine</td>
</tr>
<tr>
<td>M = # of machines at a station</td>
<td>1</td>
<td>&gt;2</td>
</tr>
<tr>
<td>D = Degree of balance</td>
<td>All E[P_{ij}] are the same (= 90 sec.)</td>
<td>All E[P_{ij}] are different</td>
</tr>
</tbody>
</table>

- MD% = Mean difference of robot operation beginning time between simulation and recursion model
- MSD% = Mean square difference of robot operation beginning time between simulation and recursion model
- Δ% = 100 (M simulation - M recursion) / (M simulation)
- ++ = Use antithetic variance reduction technique
- * = Recursion estimate, M, falls outside 95% confidence interval of fixed-sequence simulation
- ** = CPU seconds on IBM 3081-D
Figure 4.4 Relative Difference Between Recursion and Simulation Estimates of Expected Activity Beginning Time vs Activity Number (Industrial Setting).
makespan fall within applicable 95% confidence intervals; all MD% and MSD% measures are relatively small. For case 1 in Table 4.10, the relative difference between recursion and simulation estimates of expected activity beginning time, depicted in Figure 4.4, converges to a small value as the number of activities increases.

It is interesting to notice that the resulting standard deviation of makespan for all cases in Table 4.10 is relatively small. The result may be due to the large number of robot activities that were performed, the small variances of processing and robot service times (cv is 0.25) and to the fact that all lots were ready at time zero, eliminating variability in the arrival process.

4.4 Summary

Accuracy of the recursion model was evaluated relative to a number of features (e.g., lot routings, lot sequencing, parallel machines and finite buffer capacities). The recursion model gives good estimates of performance measures (e.g., makespan) compared with those derived from simulation. Furthermore, application of the recursion model in an hypothetical industrial setting demonstrates the applicability of this approach in practical production/assembly system.
CHAPTER V

MODELS FOR BINARY ASSEMBLY NETWORKS

5.1 Introduction

A four-level binary assembly network (BAN) is depicted in Figure 5.1 [see Wilhelm (1988)], which relates components required for subassembly at level four (nodes 8 - 15), subassembly operations at level three (nodes 4 - 7) and two (nodes 2 - 3), and the final product at level one (node 1). There are $(2^L - 1)$ nodes in an L-level BAN. The operation starting and finishing times for a node $i$ (i.e., a subassembly or final assembly operation) ($i=1,2,...,2^L-1$) may be described by the following two cases:

case (1): node $i$ is in the $L^{th}$ level:
\[ S_i = \text{component ready time for node } i \quad (i=2^{L-1},...,2^L-1) \]

case (2): node $i$ is in the $(L-m)^{th}$ level: ($m=1,2,...,L-1$)
\[ S_i = \max(F_{2i}, F_{2i+1}) \quad (i=2^{(L-m)-1},...,2^{(L-m)-1}) \quad (5.1) \]

In either case
\[ F_i = S_i + P_i \quad (5.2) \]

in which,

\[ L \quad = \text{total number of levels in a BAN.} \]
\[ i \quad = \text{index for node numbers in a BAN.} \]
\[ P_i \quad = \text{processing time of node } i \]
\[ S_i \quad = \text{operation starting time of node } i \]
\[ F_i \quad = \text{operation finishing time of node } i \]
Figure 5.1: A Four-level Binary Assembly Network
and nodes are numbered according to Wilhelm (1988), who shows that a BAN may be constructed to be equivalent to any arbitrary assembly tree.

In general, any distribution with support on \([0, \infty)\) may be arbitrarily closely approximated by a mixture of sufficiently many Erlang distributions of sufficiently large orders. The distribution of a random variable, \(T\), with support on \(\Omega\), a set, is defined as \(\Pr(T \in \Omega) = 1\) [Johnson and Taaffe (1988b)]. In this dissertation, the distribution of the processing time of node \(i\), \(P_i\), is assumed to be a mixture of \(n_i\) Erlang distributions, and all \(P_i\)'s are assumed to be mutually independent.

Mathematically, the distribution function of starting time, \(H_i(t)\), and of finishing time, \(Q_i(t)\), of node \(i\) may be represented as the follows:

\[
H_i(t) = \Pr(S_i \leq t) = \Pr(\max(F_{2i}, F_{2i+1}) \leq t) = \Pr(F_{2i} \leq t, F_{2i+1} \leq t) \quad (5.3)
\]

in which subscripts \(2i\) and \(2i+1\) are the numbers of the two nodes which immediately precede node \(i\). Since random variables \(F_{2i}\) and \(F_{2i+1}\) are independent, \(H_i(t)\) may be expressed as:

\[
= \Pr(F_{2i} \leq t)\Pr(F_{2i+1} \leq t) \quad (5.4)
\]

For an L-level BAN, \(Q_j(t)\), \(j=1, 2, ..., 2^L-1\), may be represented as:

\[
Q_j(t) = \Pr(F_j \leq t) = \Pr(S_j + P_j \leq t)
\]
Using the theory of conditional probability and the density function (pdf) of processing time, \( g_j(p) \) (\( p \geq 0 \)), \( Q_j(t) \) may be represented as:

\[
Q_j(t) = E[\Pr(S_j + P_j \leq t \mid P_j)] = \int_0^t \Pr(S_j + p \leq t) \cdot g_j(p) \, dp = \int_0^t g_j(p)H_j(t - p) \, dp \tag{5.5}
\]

While equations (5.3) and (5.5) give general representations, it is extremely difficult to evaluate specific distribution functions of operation starting time, \( H_i \), and operation finishing time, \( Q_i \), for node \( i \) (\( i = 1, 2, \ldots, 2^L - 1 \)) in an \( L \)-level binary assembly network in which processing times are distributed as mixtures of Erlang distributions.

The purpose of this chapter is to develop models to evaluate operation starting and finishing times in a BAN. In Section 5.2, models for approximating the distribution of operation finishing times (Model 2) are presented. In Section 5.3, bounds on the expected values of operation starting and finishing times (Model 3) are derived.

5.2 **Model 2: Approximation of the Distribution Functions of Operation Finishing Times**

Kalkunte and Wilhelm (1987a) derived formulas to obtain the distribution functions (df's) of the maximum of two independent random variables (rv's) whose distributions are mixtures of Erlang distributions and of the convolution with another rv whose distribution is also a
mixture of Erlang distributions. From equations (5.1) and (5.2), the
distribution functions of starting and finishing times of each node in the
(L-1)th level may be obtained using formulas provided in Kalkunte and
Wilhelm (1987a) for an L-level BAN if part (or component) ready-time for
each node in the Lth level is assumed to be zero. But the forms of those
derived df's are complicated and are not Erlang distributions. Hence, their
results cannot be applied to any node in the (L-m)th (m=2,3,...,L-1) level.

The approximation model developed in this dissertation uses the
exact moments of starting and finishing times of each node in the Lth
level, and approximates the df of finishing time of each of these nodes as a
mixture of Erlang distributions by matching the first three moments. The
distribution functions of finishing times of all nodes in the (L-m)th
(m=1,2,...,L-1) level may be approximated using the same concept. The
following subsections give formulas for constructing the approximation
models.

5.2.1 Moments of the Maximum of Two Independent Random Variables
and the Convolution with Another Random Variable

For convenience of discussion, we define the distribution function
of random variable $X_i$ (i=1,2,3) as a mixture of $n_i$ Erlang distributions with
mixing probability $p_{ij}$ ($j=1,2,...,n_i$), rate parameter $\mu_{ij}$ ($j=1,2,...,n_i$) and order
number $r_{ij}$ ($j=1,2,...,n_i$). Moments of starting and finishing times of each
node in the (L-m)th (m=1,2,...,L-1) level in an L-level BAN may be
obtained from the following two lemmas [Kalkunte and Wilhelm (1987a)
give similar results but different derivations], if the distribution of finishing time of each node in the \((L-m+1)\)th level is an exact (or approximated as a) mixture of Erlang distributions.

**Lemma 1:** For two independent r.v's \(X_1\) and \(X_2\), let the distribution of \(X_i\) \((i=1, 2)\) be a mixture of \(n_i\) Erlang distributions. Let \(X_{ij}\) be the \(j\)th \((j=1, 2, \ldots, n_i)\) branch of \(X_i\), the df of \(X_{ij}\) is an \(r_{ij}\) - stage Erlang distribution with rate \(\mu_{ij}\) and mixing probability \(p_{ij}\). Let \(Y_{jk} = \max(X_{1j}, X_{2k})\); the \(n\)th noncentral moment of \(Y = \max(X_1, X_2)\), \(E(Y^n)\), may be obtained as follows.

\[
E(Y^n) = \sum_{j=1}^{n_1} \sum_{k=1}^{n_2} p_{ij} p_{2k} E(Y_{jk}^n) \quad (5.6)
\]

\[
E(Y_{jk}^n) = E(X_{1j}^n) + E(X_{2k}^n) - \eta_{n}(1, j, k) - \eta_{n}(2, k, j) \quad (5.7)
\]

\[
E(X_{ij}^n) = \frac{(n + r_{ij} - 1)!}{(r_{ij} - 1)! \mu_{ij}^n} \quad (5.8)
\]

\[
\eta_{n}(i, j, k) = \frac{\mu_{ij}^{r_{ij}}}{(\mu_{ij} + \mu_{i'k})^{r_{ij} + n}} \sum_{m=0}^{r_{i'k} - 1} \frac{(n + m + r_{ij} - 1)!}{(r_{ij} - 1)! m!} \left(\frac{\mu_{i'k}}{\mu_{ij} + \mu_{i'k}}\right)^m \quad (5.9)
\]

in which \(i'\) is defined as the complement of \(i\) \((i=1, 2)\). That is, \(i'=2\) if \(i=1\), \(i'=1\) if \(i=2\).

Proof: see Appendix (1).
Lemma 2: Let $Z = Y + X_3$, the distribution of $X_3$ is a mixture of $n_3$ Erlang distributions. Let $X_{3j}$ be the $j$th ($j=1, 2, \ldots, n_3$) branch of $X_3$, the df of $X_{3j}$ is an $r_{3j}$-stage Erlang distribution with rate $\mu_{3j}$ and mixing probability $p_{3j}$. The $n$th noncentral moment of $Z$, $E(Z^n)$, may be shown to be:

$$E(Z^n) = \sum_{m=0}^{n} \frac{n!}{(n-m)!m!} E(Y^n) \left\{ \sum_{r=1}^{n_3} p_{3r} E(X_{3r}^m) \right\}$$

(5.10)

Proof:
Equation (5.10) is obtained immediately using a result from Kalkunte and Wilhelm (1987a):

$$E(Z^n) = \sum_{m=0}^{n} \frac{n!}{(n-m)!m!} E(Y^n) E(X_3^m)$$

in which the $m$th moment of $X_3$ is given by

$$E(X_3^m) = \sum_{r=1}^{n_3} p_{3r} E(X_{3r}^m).$$

5.2.2 Approximation of the Distribution Function of a Nonnegative Random Variable by a Discrete Distribution

The distribution of a nonnegative random variable is approximated by a discrete distribution in this section by invoking the properties of the Tchebycheff system (T-system) introduced by Karlin and Studden (1966). Let $T$ be a nonnegative random variable (rv) with a known upper limit $B$, such that $\Pr(0 \leq T \leq B) = 1$. It is assumed that the partial characterization of $T$ consists of the known first three noncentral moments $\mu_i$ ($i=1,2,3$) of $T$. If function $u_i(t)$ in equation (2.5) is defined as:
\[ u_i(t) = t^i \quad i = 0, 1, 2, 3, \]

the corresponding \( c_i \) \((i=1,2,3)\) in equation (2.5) is then defined as the \( i^{th} \)
moment of \( T \), that is:

\[ c_i = m_i \quad i=1, 2, 3. \]

From the properties of a T-system, if \( \{u_0(t), u_1(t), \ldots, u_4(t)\} \) (i.e.,
\( \{1, t, t^2, t^3, u_4(t)\} \)) form a T-system [i.e., satisfying inequality (2.4)], then there
exist lower and upper principal representations (pr's) of \((m_1, m_2, m_3)\). Each
of these two pr's is a discrete random variable uniquely determined by the
moments \( m_i \) \((i=1, 2, 3)\) and a prescribed number of mass points, some of
which are at prescribed locations [Karlin and Studden (1966), Eckberg
(1977)]. Table 5.1, which is a restatement of Table 1 in Eckberg (1977),
indicates the number and locations of mass points for both lower and
upper pr's with known first three moments.

The derivations for approximating the distribution of a
nonnegative \( T \), may easily be obtained from the specifications in Table
5.1. The resulting lower \( T_l \) and upper \( T_u \) of \((m_1, m_2, m_3)\) will have
the same first three moments, but will correspond to lower and upper
bounds on \( c_4 \), defined in equation (2.6), respectively. For instance, if \( u_4(t) \) is
defined as \( t^4 \), then \( T_l \) (\( T_u \)) corresponds to the lower (upper) bound on the
fourth moment of \( T \) since \( \{1, t, t^2, t^3, t^4\} \) is still a T-system. Eckberg (1977)
defined \( u_4(t) \) as \( e^{st} (s \neq 0) \) to obtain lower and upper bounds on the LST of
the df of \( T \).
Table 5.1: Number and Locations of Mass Points for the Lower and Upper Principal Representations (pr's) for Given First Three Moments.

<table>
<thead>
<tr>
<th>Lower pr</th>
<th>Upper pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>two mass points on (0,B)</td>
<td>one mass point at 0</td>
</tr>
<tr>
<td></td>
<td>one mass point on (0,B)</td>
</tr>
<tr>
<td></td>
<td>one mass point at B</td>
</tr>
</tbody>
</table>
5.2.2.1 Lower Principal Representation

From Table 5.1, it is easy to see that the lower principal representation (pr), $T_1$, has two mass points on $(0, B)$. Let $f_1(t)$ be the density function of $T_1$, $0 \leq t \leq B$, and suppose that mass points are located at $t_1$ and $t_2$, $0 < t_1 < t_2 < B$. Then, $f_1(t)$ may be represented as follows:

$$f_1(t) = \begin{cases} 
  p_1 & \text{if } t = t_1 \\
  p_1 & \text{if } t = t_2 \\
  1 - p_1 & \text{if } t = t_2 \\
  0 & \text{otherwise}
\end{cases}$$

For the given first three moments, $(m_1, m_2, m_3)$, of $T_1$, parameters (i.e., $p_1, t_1, t_2$) of the distribution of $T_1$ may be obtained by solving the following equations:

$$m_i = E(T_1^i) = p_1 t_1^i + (1 - p_1) t_2^i \quad i = 1, 2, 3 \quad (5.11)$$

Solutions to equations (5.11) are summarized in Proposition 1.

**Proposition 1:** Let $m_1, m_2, m_3$ be the first three noncentral moments of a nonnegative rv $T$. Let $T_1$, a discrete random variable, be the lower principal representation of $(m_1, m_2, m_3)$. Parameters (i.e., $p_1, t_1, t_2$) of the distribution of $T_1$ may be expressed as follows.

$$p_1 = \frac{z + x - 2m_1}{2z}, \quad t_1 = \frac{x - z}{2}, \quad t_2 = \frac{x + z}{2}$$

$$x = \frac{m_3 - m_1 m_2}{m_2 - m_1^2}, \quad y = \frac{m_2^2 - m_1 m_3}{m_2 - m_1^2}, \quad z = \sqrt{x^2 + 4y} \quad (5.12)$$

Proof: see Appendix (2).
Results of Proposition 1 agree with the lower bound on the LST of the distribution function of a nonnegative rv T with known first three moments, \((m_1, m_2, m_3)\), as discussed by Eckberg (1977, pp.138).

5.2.2.2 **Upper Principal Representation**

Table 5.1 indicates that the upper principal representation, \(T_u\), has three mass points: one at zero, one on \((0, B)\), say \(t_u\), and another at \(B\). Let \(f_u(t)\) be the density function of \(T_u\), \(0 \leq t \leq B\), then \(f_u(t)\) may be represented as follows:

\[
\begin{align*}
    f_u(t) &= \begin{cases} 
    1 - p_u - q_u & \text{if } t = 0 \\
    p_u & \text{if } t = t_u \\
    q_u & \text{if } t = B \\
    0 & \text{otherwise}
    \end{cases}
\end{align*}
\]

To facilitate implementation, the value of the upper limit, \(B\), is usually represented as a multiple of \(m_1\), say \(B = \beta m_1\). For the given first three moments \((m_1, m_2, m_3)\) of \(T_u\), parameters (i.e., \(p_u, q_u, t_u\)) of the distribution of \(T_u\) may be obtained by solving the following equations:

\[
m_i = E(T_u^i) = p_u t_u^i + q_u B^i \quad i = 1, 2, 3
\]  

(5.13)

Solutions to equations (5.13) are summarized in Proposition 2.
Proposition 2: Let \( m_1, m_2, m_3 \) be the first three noncentral moments of a nonnegative rv \( T \). Let \( T_u \), a discrete random variable, be the upper principal representation of \((m_1, m_2, m_3)\). Parameters (i.e., \( p_u, q_u, t_u \)) of the distribution of \( T_u \) may be expressed as follows.

\[
q_u = \frac{m_1 m_3 - m_2^2}{m_1 B^3 - 2m_2 B^2 + m_3 B}, \quad p_u = \frac{(m_1 - q_u B)^2}{m_2 - q_u B^2}, \quad \text{and}
\]

\[
t_u = \frac{m_2 - q_u B^2}{m_1 - q_u B}, \quad B = \beta m_1
\]

(5.14)

Proof: see Appendix (3).

Results of Proposition 2 agree with the upper bound on the LST of the distribution function of a nonnegative rv \( T \) with known first three moments, \((m_1, m_2, m_3)\), as discussed by Eckberg (1977, pp.138).

5.2.3 Approximation of the Distribution Function of a Nonnegative Random Variable by a Mixture of Erlang Distributions

In order to approximate the distribution of a nonnegative random variable as a mixture of Erlang distributions of common order, the following two lemmas are needed.

Lemma 3: Let the distribution of \( X \) be a mixture of \( n \) Erlang distributions with mixing probability \( p_j \) (\( j=1, 2, \ldots, n \)), rate parameter \( \mu_j \) (\( j=1, 2, \ldots, n \)) and common order \( r \). Let \( T \) be a discrete random variable with \( n \) mass points, \( 1/\mu_j \) (\( j=1, 2, \ldots, n \)), each with probability \( p_j \) (\( j=1, 2, \ldots, n \)). Then, moments of \( X \) and \( T \) are related as follows.
Proof:

The $k^{\text{th}}$ noncentral moment of $T$, $E(T^k)$, is given by

$$E(T^k) = \sum_{i=1}^{n} p_i \left( \frac{1}{\mu_i} \right)^k,$$

and the $k^{\text{th}}$ noncentral moment of $X_j$, whose distribution is an $r$-stage Erlang distribution with rate $\mu_j$, is given by

$$E(X_j^k) = \frac{(r+k-1)!}{(r-1)!} \mu_j^{-k}. $$

The $k^{\text{th}}$ noncentral moment of $X$, $E(X^k)$, may be represented as follows:

$$E(X^k) = \sum_{i=1}^{n} p_i \frac{(r+k-1)!}{(r-1)!} \mu_j^{-k} = \frac{(r+k-1)!}{(r-1)!} \sum_{i=1}^{n} p_i \mu_j^{-k} = \frac{(r+k-1)!}{(r-1)!} E(T^k).$$

For any nonnegative rv $X'$ with known first three moments $(m'_1, m'_2, m'_3)$, the coefficient of variation ($c$) and coefficient of skewness ($\gamma$) of rv $X'$ is defined as:

$$c = \frac{(m'_2 - m'_1^2)^{1/2}}{m'_1}, \quad \gamma = \frac{(m'_3 - 3m'_1m'_2 + 2m'_1^3)}{(m'_2 - m'_1^2)^{3/2}}.$$
In order to approximate the df of $X'$ by $G$, a mixture of Erlang distributions of common order $r$, Johnson and Taaffe (1988a) use Theorem V.10.1 in Karlin and Studden (1966) to determine the minimum order number, $r^*$, of $G$, such that $(m'_1, m'_2, m'_3)$ or $(c, \gamma)$ may be matched by $G$ [i.e., $(c, \gamma)$ is feasible for $G$]. Hence, the common order $r$ of $G$ may be obtained by setting $r$ to be greater than or equal to $r^*$, which is determined by Lemma 4 below. This is a restatement of Proposition 5 in Johnson and Taaffe (1988a).

Lemma 4: The requirement for $c$ and $\gamma$ to be feasible in a mixture of Erlang distributions of common order $r$ is that $r \geq r^*$, in which $r^*$ is determined by 

$$r^* = \max \left\{ \frac{1}{c^2}, \frac{-\gamma + 1 / c^3 + 1 / c + 2c}{\gamma - (c - 1 / c)} \right\}.$$  \hspace{1cm} (5.16)

5.2.3.1 Lower Principal Representation

Theorem 1 gives the formula for determining parameters of the lower principal representation, $X_1$, which is a mixture of two Erlang distributions of common order, to approximate the distribution of a nonnegative random variable, $X'$, by matching its first three moments.

Theorem 1: Let $m'_1$, $m'_2$ and $m'_3$ be the first three noncentral moments of rv $X'$. Let the distribution of $X_1$, the lower principal representation of $(m'_1, m'_2, m'_3)$, be a mixture of two Erlang distributions of common order $r$. The value of $r$ is determined from
Lemma 4. The parameters of the distribution of $X_1$ may be expressed as follows.

$$p_1 = \frac{z - x - 2m_1}{2z}, \quad t_1 = \frac{x - z}{2}, \quad t_2 = \frac{x + z}{2}$$

$$x = \frac{m_3 - m_1 m_2}{m_2 - m_1}, \quad y = \frac{m_2^2 - m_1 m_3}{m_2 - m_1}, \quad z = \sqrt{x^2 + 4y}$$

$$m_1 = \frac{m_1'}{r}, \quad m_2 = \frac{m_2'}{r(r + 1)}, \quad m_3 = \frac{m_3'}{r(r + 1)(r + 2)}. \quad (5.17)$$

Proof:

$m_i (i=1, 2, 3)$ is determined from Lemma 3, and parameters $p_1, t_1$ and $t_2$ are determined from Proposition 1.

Hence, the distribution of the lower pr may be represented as a mixture of two Erlang distributions: one an $r$-stage Erlang distribution with mean at $rt_1$ (or rate $1/t_1$) and mixing probability $p_1$, the other an $r$-stage Erlang distribution with mean at $rt_2$ (or rate $1/t_2$) and mixing probability $(1-p_1)$. In comparison with equations (2.7) and (2.8) [i.e., Proposition 5 in Johnson and Taaffe (1988a)], Theorem 1 gives much simpler formulas for determining the parameters of the distribution of the lower principal representation.

5.2.3.2 **Upper Principal Representation for a Random Variable With Support on [0, B]**

Theorem 2 gives the formula for determining the parameters of the upper principal representation, $X_u$, which is a mixture of three Erlang
distributions of common order, to approximate the distribution of a nonnegative random variable, $X'$, by matching its first three moments.

**Theorem 2:** Let $m'_1$, $m'_2$ and $m'_3$ be the first three noncentral moments of rv $X'$ and $\Pr(0 \leq X' \leq B)=1$. Let the distribution of $X_u$, the upper principal representation of $(m'_1, m'_2, m'_3)$, be a mixture of three Erlang distributions of common order $r$. The value of $r$ is determined from Lemma 4. The parameters of the distribution of $X_u$ may be expressed as follows.

$$q_u = \frac{m_1 m_3 - m_2^2}{m_1 B^2 - 2m_2 B^2 + m_3 B}, \quad p_u = \frac{(m_1 - q_u B)^2}{m_2 - q_u B^2}, \quad t_u = \frac{m_2 - q_u B^2}{m_1 - q_u B}$$

$$B = \beta m_1, \quad m_1 = \frac{m'_1}{r}, \quad m_2 = \frac{m'_2}{r(r+1)}, \quad m_3 = \frac{m'_3}{r(r+1)(r+2)}.$$

(5.18)

Proof:

$m_i$ ($i=1, 2, 3$) is determined from Lemma 3, and parameters $q_u, p_u$ and $t_u$ are determined from Proposition 2.

Hence, the distribution of the upper pr may be represented as a mixture of three Erlang distributions: the first is a degenerate distribution with mean at zero and mixing probability $(1-p_u-q_u)$, the second is an $r$-stage Erlang distribution with mean at $r t_u$ (or rate $1/t_u$) and mixing probability $p_u$, and the third is an $r$-stage Erlang distribution with mean at $r B$ (or rate $1/B$) and mixing probability $q_u$. Although the first branch is a degenerate distribution having mean at zero, it may be viewed as an Erlang distribution having mean at zero. Alternatively, a very large number,
\( \omega m_1 \), may be assigned to be the rate of the first branch, \( \lambda_1 \), such that \( 1/\lambda_1 \rightarrow 0 \) [Whitt (1984b)].

According to the properties of a T-system introduced by Karlin and Studden (1966), the upper principal representation always has a mass at the lower limit (e.g., zero in this section) for matching the first three moments of a nonnegative random variable. In many situations, it is not reasonable to set the lower limit to zero, for instance, processing times of assembly operations may never be zero. In the next subsection, formulas for approximating the distribution of a nonnegative random variable with lower limit \( A, A \geq 0 \), are derived.

5.2.3.3 Upper Principal Representation for a Random Variable With Support on \([A, B]\)

Following the development in Section 5.2.2.2, the upper principal representation, \( T_u \), has three mass points; one at \( A \), one on \((A, B)\), say \( t_u \), and one at \( B \). Let \( f_u(t) \) be the density function of \( T_u \) and \( \text{Pr}(A \leq T_u \leq B) = 1 \), then \( f_u(t) \) may be represented as follow:

\[
  f_u(t) = \begin{cases} 
  1 - p_u - q_u & \text{if } t = A \\
  p_u & \text{if } t = t_u \\
  q_u & \text{if } t = B \\
  0 & \text{otherwise}
\end{cases}
\]

Again, let \( A = \alpha m_1 \) and \( B = \beta m_1 \). For the given first three moments \((m_1, m_2, m_3)\) of \( T_u \), parameters (i.e., \( p_u, q_u, t_u \)) of the distribution of \( T_u \) may be obtained by solving the following equations:
\[
m_i = E(T_{x_i}) = (1 - p_u - q_u)A^i + p_u t_u^i + q_u B^i \quad i = 1, 2, 3 ,
\]
which may be expressed as:

\[
(m_i - A^i) = p_u (t_u^i - A^i) + q_u (B^i - A^i) \quad i = 1, 2, 3 .
\]

Letting \( m''_i = m_i - A^i \) and \( B'_i = B^i - A^i \) \((i=1,2,3)\), equation (5.20) may be expressed as:

\[
m''_i = p_u (t_u^i - A^i) + q_u B'_i \quad i = 1, 2, 3 .
\]

Solutions to equations (5.21) are summarized in Theorem 3.

**Theorem 3:** Let \( m'_1, m'_2 \) and \( m'_3 \) be the first three noncentral moments of rv \( X' \) and \( \Pr(A \leq X' \leq B) = 1 \). Let the distribution of \( X_u \), the upper principal representation of \((m'_1, m'_2, m'_3)\), be a mixture of three Erlang distributions of common order \( r \). The value of \( r \) is determined from Lemma 4. The parameters of the distribution of \( X_u \) may be expressed as follows.

\[
q_u = \frac{Y \pm \sqrt{Y^2 - 4XZ}}{2X} ,
\]

\[
t_u = \frac{m''_2 - q_u B'_2}{m''_1 - q_u B'_1} - A \quad , \quad p_u = \frac{m''_1 - q_u B'_1}{t_u - A} ,
\]

in which \( q_u \) is determined by subtracting the value of the radical, since adding it will give an infeasible value, and
\[ X = B_1 B_2 - B_2^2 + A B_1 B_2 - A^2 B_1^2 \]
\[ Y = m''_2 B_2 + m''_3 B_1 + A m''_2 B_1 + A m''_3 B_2 - 2A^2 m''_1 B_1 \]
\[ Z = m''_2 - m''_2 + A m''_3 + m''_2 - A^2 m''_1 \]
\[ m''_1 = m_1 - A^i, \quad B''_1 = B_1 - A^i \quad (i = 1, 2, 3) \]
\[ A = \alpha m_1, \quad B = \beta m_1, \]
and,
\[ m_1 = \frac{m'_1}{r}, \quad m_2 = \frac{m'_2}{r(r+1)}, \quad m_3 = \frac{m'_3}{r(r+1)(r+2)} \]

Proof: see Appendix (4).

In comparison with Theorem 3, Theorem 2 is a special case in which the lower limit \( A = 0 \). The distribution of the upper principal representation of a nonnegative rv with support on \([A, B], A \geq 0\), may be represented as a mixture of three Erlang distributions: the first is an \( r \)-stage Erlang distribution with mean at \( rA \) (or rate \( 1/A \)) and mixing probability \((1-p_u-q_u)/p_u\), the second is an \( r \)-stage Erlang distribution with mean at \( rt_u \) (or rate \( 1/t_u \)) and mixing probability \( p_u \), and the third is an \( r \)-stage Erlang distribution with mean at \( rB \) (or rate \( 1/B \)) and mixing probability \( q_u \).

### 5.2.4 Approximation of the Distribution Function of Operation Finishing Time by a Mixture of Erlang Distributions

In order to illustrate the procedure for approximating the distribution of operation finishing time as a mixture of Erlang distributions for each node in an \( L \)-level binary assembly network, nodes are numbered \( i = 1, 2, \ldots, 2^L - 1 \). The \( k \)th noncentral moment of the processing time of node \( i \) is denoted as \( MP(i,k) \) for \( k = 1, 2, 3 \). The \( k \)th noncentral
moment of operation starting (finishing) time of node i is designated by
MS(i,k) [ MF(i,k) ] for k=1,2,3. The approximation procedure is explicitly
stated as:

(1) Set i = 1  (i is the index of node number).
(2) Calculate MP(i, k), k= 1, 2, 3, for node i.
(3) If i = 2^{L-1}, go to Step 4.
   Else, set i = i+1 and go to Step 2.
(4) Set m = 0  [(L-m) is the level index].
(5) Set i = 2^{(L-m)-1}
   [index of the smallest node no. in the (L-m)th level].
(6) If m = 0, set MS(i, k) = the kth moment of the time at which
   part (node) i is ready to begin assembly (k= 1, 2, 3).
   Else, use MF(2i, k), MF(2i+1, k) along with Lemma 1 to estimate
   MS(i, k) for k=1, 2, 3.  [S_i = \max( F_{2i}, F_{2i+1} )].
(7) Apply Lemma 2 to calculate (or estimate) MF(i, k) for k=1, 2, 3.
   (F_i = S_i + P_i)
(8) Use the estimates of MF(i, k), k=1, 2, 3, along with Lemma 4
to determine the order number, r, required by the lower and
upper principal representations (pr's).
(9) Apply Theorem 1 (3) to obtain parameters of the distribution
of the lower (upper) pr of F_i.
   [the distribution of F_i is approximated by a mixture of two
   (three) Erlang distributions with common order r]
(10) If m = L-1, go to Step 11.
   If i < 2^{(L-m)-1}, set i = i+1 and go to Step 6.
    Else, set m = m+1 and go to Step 5.
(11) STOP.
   (approximation of the df of network completion time has
    been obtained)
5.3 **Model 3: Bounds on the Expected Values of Operation Starting and Finishing Times**

In this section, bounds on the expected values of operation starting time, $E(S_i)$, and operation finishing time, $E(F_i)$, $(i=1,2,...,2^{L-1})$ for nodes in an L-level binary assembly network are derived. New lower bounds on $E(S_i)$ and $E(F_i)$ are derived in Section 5.3.1; Devroye's (1979) lower and upper bounds on $E(S_i)$ and $E(F_i)$ are described in Section 5.3.2. Finally, Aven's (1985) upper bounds on $E(S_i)$ and $E(F_i)$ are discussed in Section 5.3.3.

### 5.3.1 New Lower Bounds

From the fundamental relationship between operation starting time of node $i$, $S_i$, ($i=1,2,...,2^{(L-1)}$) and finishing times, $F_{2i}$ and $F_{2i+1}$, [see equations (5.1) and (5.2)], the following inequality may be constructed [Devroye (1979)]:

$$
E(S_i) = E \left[ \max (F_{2i}, F_{2i+1}) \right] = E \left[ \max (S_{2i} + P_{2i}, S_{2i+1} + P_{2i+1}) \right] \geq E \{ \max \{E(S_{2i}) + P_{2i}, E(S_{2i+1}) + P_{2i+1} \} \},
$$

in which $P_{2i}$ ($P_{2i+1}$) is the processing time of node $2i$ ($2i+1$), and $P_{2i}$ and $P_{2i+1}$ are independent rv's. For convenience, let $w_1 = E(S_{2i})$, $w_2 = E(S_{2i+1})$, and assume, without loss of generality, that $w_1 \leq w_2$. The above inequality may be reformulated as:
\[
E\{ \max \left[ E(S_{2i}) + P_{2i}, E(S_{2i+1}) + P_{2i+1} \right] \} \\
= E\{ w_1 + \max \left[ P_{2i}, w_2 - w_1 + P_{2i+1} \right] \} \\
= w_1 + E\{ \max \left[ P_{2i}, w_2 - w_1 + P_{2i+1} \right] \} .
\] (5.23)

In order to compare inequality (5.23) with Lemma 1 in Section 5.2.1, let \( X_1 = P_{2i} \), \( X_2 = P_{2i+1} \) and assume the distribution of \( X_i \) \((i=1,2)\) to be a mixture of \( n_i \) Erlang distributions. Let the \( j^{th} \) \((j=1,2,...,n_i)\) branch of \( X_i \) represent rv \( X_{ij} \), an \( r_{ij} \)-stage Erlang distribution with rate \( \mu_{ij} \) and mixing probability \( p_{ij} \). Let \( Z = \max[P_{2i}, w_2 - w_1 + P_{2i+1}] \); then \( \Pr(Z \leq t), \) for \( t \geq 0 \), may be expressed as:

\[
\Pr(Z \leq t) = \Pr\{ \max \left[ P_{2i}, P_{2i+1} + w_2 - w_1 \right] \leq t \} \\
= \Pr(P_{2i} \leq t) \cdot \Pr(P_{2i+1} \leq t - w_2 + w_1) \\
= \Pr(X_1 \leq t) \cdot \Pr(X_2 \leq t - w_2 + w_1) \\
= G_1(t) \cdot G_2(t - w_2 + w_1)
\]

and

\[
\frac{d}{dt} \Pr(Z \leq t) = g_1(t)G_2(t - w_2 + w_1) + g_2(t - w_2 + w_1)G_1(t) ,
\]

in which \( g_i(t) \) and \( G_i(t) \) are the density and distribution functions of \( X_i \) \((i=1,2)\), respectively. The lower bound on \( E(S_i) \) is given by the following equation:

\[
w_1 + E\{ \max \left[ P_{2i}, P_{2i+1} + w_2 - w_1 \right] \} \\
= w_1 + E(Z) \\
= w_1 + \int_0^t \frac{d}{dt} \Pr(Z \leq t) \, dt \\
= w_1 + \int_{w_2-w_1}^{w_2} t g_1(t) G_2(t - w_2 + w_1) dt + \int_{w_2-w_1}^{w_2} t g_2(t - w_2 + w_1) G_1(t) dt .
\]
Solutions to equation (5.24) are summarized in Theorem 4.

**Theorem 4:** For two independent rv's, $X_1$ and $X_2$, let the distribution of $X_i$ ($i=1,2$) be a mixture of $n_i$ Erlang distributions. Let the $j^{th}$ ($j=1,2,...,n_i$) branch of $X_i$ represent rv $X_{jj}$ with an $r_{jj}$-stage Erlang distribution with rate $\mu_{jj}$ and mixing probability $p_{jj}$. Let $Z' = \max[Y_1+X_1, Y_2+X_2]$, in which $Y_i$ ($i=1,2$) is a nonnegative rv, $E(Y_i)=w_i$, and $w_2 \geq w_1$. A lower bound on $E(Z')$, $E(Z')$, may be expressed as:

$$E(Z') = w_1 + u_1 - u_2 + u_3 - u_4$$  \hspace{1cm} (5.25)

in which,

$$u_1 = \sum_{j=1}^{n_1} \frac{r_{1j}}{\mu_{1j}} e^{-\mu_{1j}(w_2-w_1)} \sum_{n'=0}^{r_{1j}} \frac{\mu_{1j}^{n'}(w_2-w_1)^{n'}}{n'!}$$

$$u_2 = \sum_{j=1}^{n_1} \sum_{k=1}^{n_2} p_{1j} p_{2k} \frac{r_{1j}}{(\mu_{1j} + \mu_{2k})} e^{-\mu_{1j}(w_2-w_1)} \cdot$$

$$\left( \frac{r_{1j}}{(\mu_{1j} + \mu_{2k})^{n'}} \sum_{n'=0}^{r_{1j}} \frac{(\mu_{1j} + \mu_{2k})^{n'}(w_2-w_1)^{n'}}{n'!} + \sum_{m=1}^{r_{1j}+1} \frac{\mu_{2k}^m}{(\mu_{1j} + \mu_{2k})^m} \cdot \right)$$

$$\left( \frac{r_{1j}}{(\mu_{1j} + \mu_{2k})^{n'}} \sum_{n'=0}^{r_{1j}} \frac{(\mu_{1j} + \mu_{2k})^{n'}(w_2-w_1)^{n'}}{n'!} \right)$$

$$\frac{(r_{1j}+m-1)!}{r_{1j}!(m-1)!} \right)$$
Proof: see Appendix (5).

In comparison with Theorem 4, Lemma 1 in Section 5.2.1 is a special case with \( w_1 = w_2 \). A procedure for computing the lower bounds on the expected values of operation starting, \( E(S_i) \), and finishing, \( E(F_i) \), times of node \( i \) (\( i = 1, 2, \ldots, 2^{L-1} \)) in an \( L \)-level binary assembly network is described by the following steps:

1. Set \( m = 0 \) \( (L-m) \) is the level index).
2. Set \( i = 2^{(L-m)-1} \) \( i \) is the index of node number).
3. If \( m = 0 \), set \( E(S_i) \) = the expected ready time of node \( i \).
   Else, use inequality (5.23) along with Theorem (4) to obtain \( E(S_i) \).
4. Set \( E(F_i) = E(S_i) + E(P_i) \)
5. If \( m = L-1 \), go to Step 6.
   If \( i < 2^{(L-m)-1} \), set \( i = i+1 \) and go to Step 3.
   Else, set \( m = m+1 \) and go to Step 2.
STOP (new lower bound on the expected value of network completion time has been obtained).

### 5.3.2 Devroye's Lower and Upper Bounds

Devroye (1979) developed an approach to obtain both lower and upper bounds on the mean and an upper bound on the variance of completion time of a PERT network by using only the means and variances of individual activity durations. Since Devroye's method does not require the distribution functions of activity durations to be known, it may be applied to the binary assembly network (BAN) in which the distributions of processing times of nodes in this BAN are mixtures of Erlang distributions. Devroye's bounds are defined as follows.

(A) The lower bounds on the expected values of starting, $E(S_i)$, and finishing, $E(F_i)$, times of node $i$ ($i=1, 2, ..., 2^{L-1}$) in an $L$-level BAN:

1. Set $m = 0$ [(L-m) is the level index].
2. Set $i = 2^{(L-m)-1}$ (i is the index of node number).
3. If $m = 0$, set $E(S_i) = $ the expected ready time of node $i$.
   Else, set $E(S_i) = \max \{E(F_{2i}), E(F_{2i+1})\}$.
4. Set $E(F_i) = E(S_i) + E(P_i)$
5. If $m = L-1$, go to Step 6.
   If $i < 2^{(L-m)-1}$, set $i = i+1$ and go to Step 3.
   Else, set $m = m+1$ and go to Step 2.
6. STOP (Devroye's lower bound on the expected value of network completion time has been obtained).
(B) The upper bounds on the means of starting, $E(S_i)$, and finishing, $E(F_i)$, times and the variances of starting, $\sigma^2(S_i)$, and finishing, $\sigma^2(F_i)$, times of node $i$ ($i=1, 2, ..., 2^L-1$) in an $L$-level BAN:

1. Set $m = 0$ [(L-m) is the level index].
2. Set $i = 2^{(L-m)-1}$ (i is the index of node number).
3. If $m = 0$, set
   
   $E(S_i) = \text{the mean of the ready time of node } i$,  
   $\sigma^2(S_i) = \text{the variance of the ready time of node } i$.

   Else, set
   
   $E(S_i) = \max[E(F_{2^m}), E(F_{2^{m+1}})] + \left(2 \max[\sigma^2(F_{2^m}), \sigma^2(F_{2^{m+1}})]\right)^{1/2}$
   $\sigma^2(S_i) = \sigma^2(F_{2^m}) + \sigma^2(F_{2^{m+1}})$

4. Set $E(F_i) = E(S_i) + E(F_i)$,
   $\sigma^2(F_i) = \sigma^2(S_i) + \sigma^2(F_i)$.

5. If $m = L-1$, go to Step 6.
   If $i < 2^{(L-m)-1}$, set $i = i+1$ and go to Step 3.
   Else, set $m = m+1$ and go to Step 2.

6. STOP (Devroye's upper bounds on the mean and variance of network completion time has been obtained).

Since Devroye's method only considers the means and variances of processing times, $E(S_i)$, $E(F_i)$, $E(S_i)$, $E(F_i)$, $\sigma^2(S_i)$ and $\sigma^2(F_i)$ obtained from his method may not be too tight for a BAN in which the distributions of processing times are mixtures of Erlang distributions. It is expected, however, that Devroye's bounds may perform better if processing times are normally distributed, since the normal distribution is determined by its mean and variance.
5.3.2 Aven's Upper Bounds

Aven (1985) developed two different formulas that give upper bounds on the mean of the maximum of a number of random variables using only the means and variances of those rv's. He claimed that his upper bound is tighter than Devroye's. Since Aven (1985) developed bounds just for the mean, but not the variance of the maximum of a set of rv's, his method can only be applied to obtain an upper bound on the mean starting and finishing times of nodes in the (L-1)th level of an L-level BAN. However, if Devroye's upper bound on the variance is incorporated, the upper bounds on the means and variances of starting and finishing times of nodes in any level of a BAN may be obtained. Aven's two formulas specialized to analyze a binary assembly network are presented in Lemma 5.

Lemma 5: For two independent random variables $X_1$ and $X_2$, let $Z = \max\{X_1, X_2\}$. Two different upper bounds on $E[Z]$, denoted as $E_1(Z)$ and $E_2(Z)$, may be obtained as follows.

\[
E_1(Z) = \bar{m} + \left(\sigma_0^2 + \bar{\mu} - \bar{m}^2\right)^{1/2}
\]

\[
E_2(Z) = \max\left[E(X_1), E(X_2)\right] + \left(\sigma_0^2\right)^{1/2}
\]

in which,

\[
\bar{m} = \frac{E(X_1) + E(X_2)}{2}, \quad \bar{\mu} = \frac{E^2(X_1) + E^2(X_2)}{2}, \quad \sigma_0^2 = \frac{\sigma^2(X_1) + \sigma^2(X_2)}{2}.
\]

Proof: immediate from Theorem 2.1 in Aven (1985).
It is to be noted that Aven's two formulas give the same upper bounds on the expected value of $\max[X_1, X_2]$ if $X_1$ and $X_2$ are independent and identically distributed. The resulting upper bound is $E(X_i) + [\sigma^2(X_i)]^{1/2}$ $(i=1 \text{ or } 2)$. The upper bounds on the means of starting, $E_k(S_i)$, and finishing, $E_k(F_i)$, times and the variances of starting, $\sigma_k^2(S_i)$, and finishing, $\sigma_k^2(F_i)$, times of node $i$ $(i=1,2,...,2^L-1)$ in an $L$-level BAN using Aven's (1985) formula $k$ $(k=1,2)$ may be calculated as follows.

1. Set $m = 0$ [(L-m) is the level index].
2. Set $i = 2(L-m)-1$ (i is the index of node number).
3. If $m = 0$, set
   $E_k(S_i) = \text{the mean of the ready time of node } i$,
   $\sigma_k^2(S_i) = \text{the variance of the ready time of node } i$.
   Else, apply Lemma 5 to obtain $E_k(S_i)$ \[S_i = \max(F_{2i}, F_{2i+1})\]
   $\sigma_k^2(S_i) = \sigma_k^2(F_{2i}) + \sigma_k^2(F_{2i+1})$
4. Set $E_k(F_i) = E_k(S_i) + E(P_i)$
   Set $\sigma_k^2(F_i) = \sigma_k^2(S_i) + \sigma^2(P_i)$
5. If $m = L-1$, go to Step 6.
   If $i < 2(L-m)-1$, set $i = i+1$ and go to Step 3.
   Else, set $m = m+1$ and go to Step 2.
6. STOP (Aven's upper bounds on the mean and variance of network completion time has been obtained).
5.4 **Summary**

Model 2 gives approximations (i.e., the lower and upper pr's) of the distribution of any nonnegative random variable as a mixture of two (or three) Erlang distributions of common order. Algorithms based on the lower and upper pr's were developed to approximate the df's of operation finishing times of nodes in a BAN. Model 3 gives a new lower bound on the expected values of operation starting and finishing times of nodes in a BAN; Devroye's and Aven's methods were used to obtain bounds on the means and variances of operation starting and finishing times using only the means and variances of the processing times.
CHAPTER VI
NUMERICAL EVALUATIONS OF MODELS 2 AND 3

6.1 Introduction

In this chapter Models 2 and 3 are evaluated from several perspectives. In Section 6.2, approximations of the df of any nonnegative random variable with known first three moments obtained from the lower and upper pr's are compared with those obtained from the method of Johnson and Taaffe (1988a). Then, the accuracy of Model 2 (i.e., the df approximations of operation finishing times of nodes in a BAN) with different types of processing time distributions is evaluated in Section 6.3. Finally, the tightness of Model 3 (i.e., bounds on the expected values of operation starting and finishing times of nodes in a BAN) with different types of processing time distributions is evaluated in Section 6.4.

6.2 Approximation Methods for Matching the First Three Moments of a Nonnegative Distribution

Johnson and Taaffe (1988a and 1988b) evaluate the acceptability of the distribution approximations determined by their method [i.e., equations (2.7) and (2.8) in Section 2.6] in terms of dimension, numerical stability, kurtosis, structure of the distribution representation, and density function shape. The lower pr developed in this dissertation is equivalent to the method of Johnson and Taaffe (1988a) which is based on a complicated derivation of the three moments matching problems that
expresses results in totally different forms, the reader may review Johnson and Taaffe (1988a and 1988b) for analysis of their approximation.

Some important properties of the df approximation, a mixture of two Erlangs with mixing probability \( p_i \) (i=1,2), rate parameter \( \lambda_i \) (i=1,2) and common order \( r \), determined by the method of Johnson and Taaffe (1988a) are summarized as follows [Johnson and Taaffe (1988b)]:

(a) Consider the shape of the density function; large values of \( \chi (=\lambda_2/\lambda_1, \lambda_2 \geq \lambda_1) \), \( p_1 \) or \( p_2 \), and \( r \) tend to induce bimodality, while small values of \( \chi \) and \( r \) help to maintain unimodality. For \( g \), a density function of a nonnegative rv, point \( x' \) is a mode of \( g \) if and only if there exists an \( \epsilon > 0 \) such that \( g(x') \geq g(x) \) for \( x \in (x'-\epsilon, x'+\epsilon) \). Zero is a mode of \( g \) if and only if \( g(x_1) \geq g(x_2) \) for \( 0 < x_1 < x_2 < \epsilon \).

(b) By approximating some known distribution (e.g., the Weibull, gamma or lognormal distribution) as a mixture of two Erlang distributions of common order, the density-function shapes are not good approximations for the Weibull and gamma (with coefficient of variation greater than one) distributions.

From property (a), the approximating density functions, a mixture of two Erlang distributions, obtained from Johnson and Taaffe (1988a) and the lower pr methods cannot be unimodal if the approximating distributions have extreme values of \( \chi \) (e.g., \( \chi = 50 \)) and \( p_1 \) (e.g., \( p_1 \geq 0.9 \) or \( \leq 0.1 \)). In this section, one of the cases with extreme values of \( \chi \) and \( p_1 \) considered by Johnson and Taaffe (1988b) is used to show that the approximating density function obtained from the upper pr may attain unimodality. Examples used to approximate Weibull and gamma
distributions in Johnson and Taaffe (1988b) are also analyzed using the upper pr developed in this dissertation.

In Johnson and Taaffe (1988b), all density functions are scaled to have unit mean and the parameters of a distribution are represented as a point \((c-1/c, \gamma)\) on the feasible region of a distribution which was discussed in Johnson and Taaffe (1988a). Given the unit mean (i.e., \(m_1=1\)), coefficient of variation \((c)\) and coefficient of skewness \((\gamma)\), the second and third moments may be obtained as follows

\[
m_2 = c^2 + 1, \\
m_3 = \gamma (m_2 - 1)^{3/2} + 3m_2 - 2.
\]

The lower and upper pr's can then be used to approximate the distribution with known first three moments \((1, m_2, m_3)\). Due to the difficulty of determining the shape and scale parameters of a Weibull distribution with known first three moments \((m_1, m_2, m_3)\), the original graph of the Weibull density functions in Johnson and Taaffe (1988b) are used to compare the accuracy of the approximating density functions obtained from the lower and upper pr's in the examples of approximating Weibull distributions.

Starting with an unknown distribution with \((c-1/c, \gamma)=(-1.175, 0.415)\), the shape of the approximating density function, with \(\chi=50\) and \(p_1=.9\), obtained from the lower pr is bimodal (see Figure 6.1), and that from the upper pr (setting of the lower limit \(A=0.1\)) is also bimodal with a sharp peak at 0.1 (see Figure 6.1). But, a smooth and unimodal density function
Figure 6.1 PDF's of the Lower and Upper (A=0.1) pr's with 
(c-1/c, γ) = (-1.175, .415).
can be obtained from the upper pr, simply by setting $A=0.0$ and not changing any other parameters (see Figure 6.2).

A Weibull distribution with $(c-1/c, \gamma) = (-4.967, -0.373)$ and density $f(t)$, $t \geq 0$, has one mode and is slightly skewed to the left (see Figure 6.3). The approximating density function obtained from the lower pr has two modes; each represents an (almost symmetric) Erlang density function of order 32 (see Figure 6.3 and 6.4). By setting the lower limit $A=0.0$, the density function approximated by the upper pr has only one mode (see Figure 6.4).

Next, consider a Weibull distribution with $(c-1/c, \gamma) = (-2.284, 0.206)$. The Weibull density $f(t)$, $t \geq 0$, has one mode and is almost symmetric (see Figure 6.5). The approximating density function obtained from the lower pr has two modes and does not fit well (see Figure 6.5 and 6.6). The density function approximated by the upper pr with $A=0.0$ has only one mode and gives a better fit to the Weibull density; however, it still overestimates $f(t)$ for $t$ near 1.0 (see Figure 6.6).

Now, consider another Weibull distribution with $(c-1/c, \gamma) = (-0.978, 0.917)$. The Weibull density $f(t)$, $t \geq 0$, has one mode and is skewed to the right (see Figure 6.7). Again, the approximating density function obtained from the upper pr with $A=0.0$ has one mode and gives a better fit to the Weibull density, except that it underestimates $f(t)$ for $t$ less than 1.0 (see Figure 6.7 and 6.8).

Since the Erlang distribution is a special case of the gamma distribution, it is expected that a gamma distribution with $c \leq 1$ may be well approximated by a mixture of Erlang distributions [Johnson and Taaffe (1988b)]. For a gamma distribution with $(c-1/c, \gamma) = (-1.658, 0.940)$ [i.e.,
Figure 6.2 PDF's of the Lower and Upper (A=0.0) pr's with 
\((c-1/c, \gamma)=(-1.175, .415)\).
Figure 6.3  PDF's of Weibull Distribution Compared with Johnson and Taaffe's Approximation with $(c-1/c, \gamma)=(-4.967, -.373)$. 
Figure 6.4 PDF's of the Lower and Upper (A=0.0) pr's with (c-1/c, γ)=(-4.967, -.373).
Figure 6.5 PDF's of Weibull Distribution Compared with Johnson and Taaffe's Approximation with \((c-1/c, \gamma)=(-2.284, .206)\).
Figure 6.6 PDF's of the Lower and Upper (A=0.0) pr's with (c-1/c, γ)=(-2.284, .206).
Figure 6.7 PDF's of Weibull Distribution Compared with Johnson and Taaffe's Approximation with \((c-1/c, \gamma)=(-.978, .917)\).
Figure 6.8  PDF's of the Lower and Upper \( (A=0.0) \) pr's with \((c-1/c, \gamma)=(-.978, .917)\).
c=0.470], the approximating density functions obtained from both lower and upper, (setting A=0.0 ), pr's (see Figure 6.10) match the density function of the gamma distribution f(t), t ≥ 0, almost exactly (see Figure 6.9) with the lower pr giving the better of the two fits.

For a gamma distribution with (c-1/c, γ) = (1.0, 3.236) [i.e., c=1.618], the c is greater than 1, the gamma density function f(t) (see Figure 6.11), t ≥ 0, was poorly approximated by the lower pr (see Figure 6.11 and 6.12). However, the upper pr (setting A=0.1) gives a very good approximation of f(t) for t > 0 (see Figure 6.12).

In general, a unimodal (bimodal) approximating density function, in which c > 1, can be obtained from the upper pr by simply setting the lower limit A to be zero (a positive real number). Neither lower or upper pr can approximate the density function of the Weibull distribution very well. The upper pr can approximate the density function of the gamma distribution very well, but the lower pr can give good approximations only with c≤1.

6.3 Accuracy of Model 2

In this section, the accuracies of Model 2 (i.e., the df approximations of operation finishing times in a binary assembly network) with different types of processing time distributions (e.g., mixtures of Erlangs, exponential and normal distributions) are evaluated. It is assumed, without loss of generality, that all components in the last level L are ready to begin assembly at time 0.
Figure 6.9 PDF's of Gamma Distribution Compared with Johnson and Taaffe's Approximation with \((c-1/c, \gamma) = (-1.658, .940)\).
Figure 6.10 PDF's of the Lower and Upper ($A=0.0$) pr's with $(c-1/c, \gamma)=(-1.658, .940)$
Figure 6.11 PDF's of Gamma Distribution Compared with Johnson and Taaffe's Approximation with (c-1/c, γ)=(1.0, 3.236).
Figure 6.12 PDF’s of the Lower and Upper (A=0.1) pr’s with 
(c−1/c, γ)=(1.0, 3.236).
6.3.1 Case 1: Processing Times are Distributed as Mixtures of Erlangs

Test #1 (a). The first test was designed to evaluate the accuracy of Model 2 in a four-level, balanced BAN, in which all nodes have the same processing time distribution, a mixture of two Erlang distributions with mixing probability \( p_i \) (i=1,2), rate parameter \( \lambda_i \) (i=1,2) and order number \( n_i \) (i=1,2).

From Section 6.2, we know that the lower pr cannot produce good (e.g., smooth and unimodal) density function shapes for extreme values of \( \chi = \lambda_2/\lambda_1, \lambda_2 \geq \lambda_1 \) and \( p_1 \). In this numerical example, we consider \( \chi \) and \( p_1 \) as factors with two different levels (see Table 6.1) and set \( n_1=3, n_2=4 \). Hence, four runs were made in total. For the case of a balanced BAN, expected starting (finishing) times of all nodes at a given level are the same. Approximations of the first three moments of operation finishing times of all nodes in each level determined by the lower and upper pr's are listed in Table 6.2.

Table 6.2 shows that, for all four runs, the first three noncentral moments of operation finishing times of all nodes in level three obtained from the lower pr are the same as those from the upper pr. That is because the moments of starting times at level three are determined exactly [using equations (5.6) - (5.9)]. For nodes in levels one and two, moments of the lower and upper pr's are very close, but moments obtained from the lower pr are slightly smaller than those from the upper pr.
Table 6.1: Values of $\chi$ and $p_1$ at Two Different Levels.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi = \lambda_2/\lambda_1$</td>
<td>$2 \left( \begin{array}{l} \lambda_1 = 0.8 \ \lambda_2 = 1.6 \end{array} \right)$</td>
</tr>
<tr>
<td>$p_1$</td>
<td>0.4</td>
</tr>
</tbody>
</table>
Table 6.2: Moments of Operation Finishing Times of Nodes in a Four-Level, Balanced BAN with Case 1 Settings.

<table>
<thead>
<tr>
<th>RUN Factors Level</th>
<th>Lower PR</th>
<th>Upper PR</th>
<th>New Devroye's Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M_1</td>
<td>M_2</td>
<td>M_3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>l</td>
<td>p_i</td>
<td>M_1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8.059</td>
<td>75.419</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.620</td>
<td>7.530</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.625</td>
<td>17.656</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6.937</td>
<td>54.849</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.000</td>
<td>12.188</td>
</tr>
</tbody>
</table>

Average Run Time*: 0.3602

Devroye's Upper Bound:

M_k = the k* noocentral moment of operation finishing time (k=1,2,3)

* CPU seconds on IBM 3081-D
<table>
<thead>
<tr>
<th>RUN</th>
<th>Factors</th>
<th>Simulation+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RUN</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AHW% = 100 (half-width of the 99.5% confidence interval/mean)
+ = # of replications = 100,000
Since no analytical model is available to evaluate the accuracy of Model 2 with Case 1 settings, simulation model is developed to construct 99.5% confidence intervals for the expected values of operation finishing times and the percentage of relative half width (AHW%), defined as the ratio of the half width of the confidence interval to the estimated mean, is controlled to be less than 0.65. Three (None) of four approximated mean network completion times (i.e., operation finishing time of node 1 at level 1) in the four runs obtained from the lower (upper) pr fall within the 99.5% confidence interval. The approximated variances of operation finishing times obtained from both pr's are relatively close to the simulation estimated variances with the lower pr giving the closer of the two approximations.

In order to study the distribution of network completion time as approximated by the lower and upper pr's, approximating density functions and parameters of the distributions of network completion time from run (1) and (2) are presented in Figure 6.13, 6.14 and Table 6.3, respectively. In all figures of tests used to evaluate the accuracy of Model 2, 'Δ' and '*' represent the approximated density (or distribution) functions of operation finishing times evaluated at a specific time using the lower and upper pr's, respectively. Points of '*' are connected by a solid line to illustrate the shape of the approximated density (or distribution) functions obtained from the upper pr.

For both runs in Table 6.3, the mixing probability of the second branch in the mixture of three Erlang distributions obtained for the upper pr is very large. That is, the second Erlang distribution in an upper pr
Table 6.3: Parameters of the df of Network Completion Time Approximated by the Lower and Upper PR’s in Test # 1 (a).

<table>
<thead>
<tr>
<th>RUN</th>
<th>Factors</th>
<th>PR</th>
<th>Common Order</th>
<th>Mixing Probability</th>
<th>Rates at Each Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( r ) ( p_1 ) ( p_2 ) ( p_3 ) ( \mu_1 ) ( \mu_2 ) ( \mu_3 )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>0.9</td>
<td>LOWER</td>
<td>22</td>
<td>0.4880 0.5120 -- 1.2498 1.1047 --</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UPPER</td>
<td>22</td>
<td>0.819E-3 0.9990 0.140E-3 2359.9582 1.1797 0.2360</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>0.4</td>
<td>LOWER</td>
<td>9</td>
<td>0.1246 0.875 -- 1.3353 0.7291 --</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UPPER</td>
<td>9</td>
<td>0.1009E-1 0.9894 0.549E-3 1564.3328 0.7760 0.1564</td>
</tr>
</tbody>
</table>
Figure 6.13 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Run 1 of Test #1 (a).
Figure 6.14 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Run 2 of Test #1 (a).
dominates the entire mixture of Erlang distributions. The density functions obtained from both pr's are almost identical for run (1) (see Figure 6.13). The density function obtained from the upper pr in run (2) has a higher peak around the mean (see Figure 6.14).

Figures 6.15 and 6.16 show the corresponding approximating distribution functions from run (1) and (2), respectively. Again, the df's of both pr's in run (1) are almost identical (see Figure 6.15), but there are remarkable differences between these two df's in run (2) for small values of t (i.e., t ≤ 3). The df obtained from the upper pr in run (2) jumps from 0.0 to 0.0101 when t changes from zero to the recorded t nearest zero (e.g., 0.3835 in Figure 6.16), but the df obtained from the lower pr remains at zero for 0 ≤ t ≤ 0.3835.

This difference is due to the fact that the distribution approximated by the upper pr is always a mixture of three Erlang distributions, and the first branch is a degenerate distribution with mean at zero. This degenerate distribution is modeled as an Erlang distribution by assigning it a very large rate parameter μ₁. The df, F(t) for t ≥ 0, of a mixture of three Erlang distributions with common order r, mixing probability pᵢ (i=1, 2, 3) and rate parameter μᵢ (i=1, 2, 3) is defined as:

\[ F(t) = 1 - \sum_{i=1}^{3} p_i e^{-\mu_i t} \sum_{m=0}^{r-1} \frac{(\mu_i t)^m}{m!} \]

\[ = 1 - \sum_{i=1}^{3} p_i \sum_{m=0}^{r-1} \frac{1}{m!} \frac{(\mu_i t)^m}{e^{\mu_i t}} \]

\[ = 1 - \sum_{i=1}^{3} p_i R_i \]
Figure 6.15 DF of Network Completion Time Approximated by the Lower and Upper pr's in Run 1 of Test #1 (a).
Figure 6.16 DF of Network Completion Time Approximated by the Lower and Upper pr's in Run 2 of Test #1 (a).
The value of \( R_i \) is zero when \( \mu_i \) (i=1, 2, 3) is very large and \( t \) is small, [i.e., \( t \leq \varepsilon \) (\( \varepsilon > 0 \))]. Large values of \( \mu_i \) have a small effect when the mixing probability \( p_i \) is small [e.g., \( p_1=0.819E-3 \) in run (1) of Table 6.3], because the entire mixing distribution will depend heavily on the other two branches, and the value of \( F(t) \) will be close to zero when \( t \leq \varepsilon \) (see Figure 6.15). The effect of a large \( \mu_i \) becomes obvious when \( p_i \) is not negligible [e.g., \( p_1=0.1009E-1 \) in run (2) of Table 6.3], the value of \( F(t) \), \( t \leq \varepsilon \), will then be slightly greater than zero (see Figure 6.16). One way to overcome this disadvantage of the upper \( pr \) is simply to replace the distribution function involving jumps by the smooth df obtained from the lower \( pr \), because the density function obtained from the upper \( pr \) over this interval is less than or equal to the density function from the lower \( pr \).

For approximations, it is interesting to know the relative difference \( (RD_i) \) of the \( i^{th} \) moment of network completion time as estimated by two different methods (e.g., the lower and upper \( pr \)'s in Table 6.2); \( RD_i \) is defined by

\[
RD_i = \frac{m'_i - m''_i}{m''_i}
\]

in which,

\( m'_i = \) the \( i^{th} \) moment of network completion time obtained (or approximated) from method 1,

\( m''_i = \) the \( i^{th} \) moment of network completion time obtained (or approximated) from method 2.

Considering the four runs described by Table 6.2, run (2) has the largest \( RD_1 \) (=1.165%) between the lower and upper \( pr \)'s, because of the
non-negligible mixing probability, $p_1 = 0.1009E-1$, of the first (degenerate) branch of the df of the upper pr. Table 6.2 also shows that the average runtime over the four runs in Test #1(a) for the lower (upper) pr is 0.3602 (0.6285) CPU seconds, the runtime required by the upper pr is greater than that by the lower pr, because of the longer runtime required to estimating moments of operation starting times [see equation (5.6) in Lemma 1] by the upper pr.

In Test #1(a), there are only 15 nodes in the BAN. It is interesting to consider another, more complicated, eight-level BAN in which there are 255 nodes.

Test #1 (b). This test evaluated the accuracy of Model 2 in an eight-level, balanced BAN with the same parameter settings used in Test #1(a).

For the eight-level, balanced BAN with the same parameter settings used in run (2) of Test #1(a), approximations of the first three moments of finishing times of nodes in each level obtained from the lower and upper pr's are listed in Table 6.4. Again, the first two moments obtained from both pr's are very close, even with many more nodes. However, RD3 between the lower and upper pr's increases with the number of levels. For instance, RD3 is 2.095% in the four-level BAN and is 8.385% in the eight-level BAN. The runtime of run (2) in Test #1(b) ($\chi=25$, $p_1=0.4$) for the lower (upper) pr is 8.8596 (12.8189) CPU seconds.

In comparison with simulation estimates, approximated mean operation finishing times for nodes in the last seven (three) levels obtained from the lower (upper) pr fall within the 99.5% confidence
Table 6.4: Moments of Operation Finishing Times of Nodes in an Eight-Level, Balanced BAN with Case 1 Settings ($x = 25$, $p_1 = 0.4$).

<table>
<thead>
<tr>
<th>Level</th>
<th>Lower PR $M_1$</th>
<th>Upper PR $M_1$</th>
<th>New $M_1$</th>
<th>Devroye's L.B.</th>
<th>Aven's U.B.</th>
<th>Devroye's U.B.</th>
<th>Upper Bound on $M_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$M_2$</td>
<td>$M_3$</td>
<td>$M_2$</td>
<td>$M_3$</td>
<td>$M_2$</td>
<td>$M_3$</td>
<td>$M_2$</td>
</tr>
<tr>
<td>1</td>
<td>28.511</td>
<td>838.021</td>
<td>25428.856</td>
<td>27.840</td>
<td>797.186</td>
<td>23461.495</td>
<td>20.375</td>
</tr>
<tr>
<td>3</td>
<td>19.868</td>
<td>417.600</td>
<td>9269.142</td>
<td>19.345</td>
<td>396.221</td>
<td>8566.421</td>
<td>15.016</td>
</tr>
<tr>
<td>8</td>
<td>1.620</td>
<td>7.530</td>
<td>46.884</td>
<td>1.620</td>
<td>7.530</td>
<td>46.884</td>
<td>--</td>
</tr>
</tbody>
</table>

$M_k$ = the $k$th noncentral moment of operation finishing time ($k = 1, 2, 3$).

* = CPU seconds on IBM 3081-D.
Table 6.4 Continued:

<table>
<thead>
<tr>
<th>Level</th>
<th>Mean</th>
<th>Var.</th>
<th>L.B.</th>
<th>U.B.</th>
<th>AHW%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.60</td>
<td>26.29</td>
<td>28.55</td>
<td>28.65</td>
<td>0.163</td>
</tr>
<tr>
<td>2</td>
<td>24.19</td>
<td>25.06</td>
<td>24.15</td>
<td>24.24</td>
<td>0.188</td>
</tr>
<tr>
<td>3</td>
<td>19.86</td>
<td>23.31</td>
<td>19.82</td>
<td>19.91</td>
<td>0.221</td>
</tr>
<tr>
<td>4</td>
<td>15.66</td>
<td>21.47</td>
<td>15.62</td>
<td>15.71</td>
<td>0.269</td>
</tr>
<tr>
<td>5</td>
<td>11.60</td>
<td>19.07</td>
<td>11.56</td>
<td>11.64</td>
<td>0.343</td>
</tr>
<tr>
<td>6</td>
<td>7.78</td>
<td>16.04</td>
<td>7.74</td>
<td>7.82</td>
<td>0.469</td>
</tr>
<tr>
<td>7</td>
<td>4.32</td>
<td>11.54</td>
<td>4.29</td>
<td>4.35</td>
<td>0.715</td>
</tr>
<tr>
<td>8</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

AHW% = 100 (half-width of the 99.5% confidence interval/mean)
+ = # of replications = 80,000
* = CPU seconds on IBM 3081-D = 2509.63
intervals with relatively small AHW% (see Table 6.4). In general, the lower pr gives a better approximations of the means and variances of operation finishing times than the upper pr in Test # 1 (a) and (b).

In order to compare accuracies of approximations in four and eight-level BAN's, the approximating density functions and parameters of the distributions of network completion time for runs (1) and (2) are presented in Figure 6.17, 6.18 and Table 6.5, respectively. From run (1) of Table 6.5, it can be seen that the mixing distribution obtained from the lower (upper) pr is determined by the first (second) Erlang distribution of order 80 (83) and rate parameter 1.8877 (1.9729). With such a high order Erlang distribution, the shape of the density functions obtained from both pr's are similar to a normal distribution with a relatively small variance (see Figure 6.17). A similar situation occurs in run (2) of Table 6.5 as shown in Figure 6.18. Actually, the result of run (1) in Table 6.5 and Figure 6.17 is consistent with the assumption of normally distributed processing times, an assumption invoked by earlier recursion models applied to assembly systems (e.g., see Chapter 3).

6.3.2 Case 2: Exponentially Distributed Processing Times

Test #2. The second test evaluated the accuracy of Model 2 in a three-level, imbalanced BAN with independent and exponentially distributed node processing times.

Kulkarni and Adlakha (1986) modeled a PERT network with independent and exponentially distributed activity durations as a finite-
Table 6.5: Parameters of the df of Network Completion Time Approximated by the Lower and Upper PR's in Test # 1 (b).

<table>
<thead>
<tr>
<th>RUN</th>
<th>Factors</th>
<th>PR</th>
<th>Common Order</th>
<th>Mixing Probability</th>
<th>Rates at Each Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>χ</td>
<td>p₁</td>
<td>r</td>
<td>p₁</td>
<td>p₂</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>0.9</td>
<td>LOWER 80</td>
<td>0.99999</td>
<td>0.1E-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UPPER 83</td>
<td>0.6E-4</td>
<td>0.99993</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>0.4</td>
<td>LOWER 34</td>
<td>0.99925</td>
<td>0.75E-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UPPER 35</td>
<td>0.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Figure 6.17 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Run 1 of Test #1 (b).
Figure 6.18 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Run 2 of Test #1 (b).
state, absorbing, continuous-time Markov chain with upper triangular generator matrix. In order to illustrate the relationships between BAN and PERT networks and demonstrate the flexibility of Model 2, the exact distribution function of network completion time obtained using their method is compared with the approximations of both pr's.

Since the method of Kulkarni and Adlakha requires that a network have a single source node and a single sink node, the three-level BAN can be recast as in Figure 6.19. Dotted lines in Figure 6.19 represent 'dummy' activities; a node (e.g., node 2) at level two can start operation immediately after its preceding node (e.g., node 4) finishes. The BAN in Figure 6.19 can be converted to an equivalent PERT network, depicted in Figure 6.20, that can be modeled by the method of Kulkarni and Adlakha. In Figure 6.20, $\lambda_i$ (i=1,2,...,5) indicates the rate parameter of activity i. It is assumed that $\lambda_1=1.0$, $\lambda_2=1.2$, $\lambda_3=1.5$, $\lambda_4=0.8$ and $\lambda_5=1.2$ in this numerical example.

The basic requirements of the method of Kulkarni and Adlakha are that the state space and transition rates be determined, so that the infinitesimal generator matrix can be constructed. Although Kulkarni and Adlakha (1986) discussed a method to determine state space and transition rates, the Order-of-Processing (OP) diagram introduced by Fisher et al. (1985) seems to be more efficient. The OP diagram of Figure 6.20 is depicted in Figure 6.21.

Nodes in Figure 6.21 represent states. Each $U_j$ (j=0,1,...,9) corresponding to a node in the OP diagram is defined by the set of activities currently executing, which are indicated in the upper half of a block, and the set of activities that have completed execution, which are
Figure 6.19 A Three-level BAN Consisting of Dummy Activities.

Figure 6.20 The Single Source/Single Sink PERT Network which Corresponds to Figure 6.19.
Figure 6.21 The Order-of-Processing (OP) Diagram which Corresponds to Figure 6.20.
indicated in the lower half of a block. An arrow along with \( \lambda_i \) (i=1,2,..,5) which connects two different states indicates the transition rate between these two states. With the information listed in Figure 6.21, the infinitesimal generator matrix, \( Q \), may be determined easily.

Moments of network completion time obtained from Kulkarni and Adlakha's backward algorithm and approximated by both lower and upper pr's are listed in Table 6.6. From Table 6.6, the first three moments of network completion time approximated by both pr's are very close to the exact results and the absolute value of average RD\(_i\) (i=1,2,3) for the first three moments between the lower (upper) pr and Kulkarni and Adlakha's analytical result is 0.104% (0.631%). The CPU time for the method of Kulkarni and Adlakha is 1.7054 seconds whereas the runtime for the lower (upper) pr is only 0.0543 (0.0882) seconds.

Although the analytical expression for the df of network completion time, \( F(t) \), [with exponentially distributed activity times] is derived by Kulkarni and Adlakha (1986), they also give formulas to bound \( F(t) \) from below and above within \( \epsilon \) for numerical computing purpose. In this three-level BAN, \( \epsilon \) is set to equal 10\(^{-5}\), so that the resulting lower and upper bounds on \( F(t) \), for \( t \geq 0 \), will have five place accuracy. \( F(t) \) obtained from Kulkarni and Adlakha's method are compared with those from the lower and upper pr's are given in Table 6.7, Figure 6.22, and 6.23.

Data in Table 6.7 indicates that the values of \( F(t) \) obtained from the three methods are very close, and the shape of the df's obtained from the method of Kulkarni and Adlakha and the lower pr are almost identical (see Figure 6.22 and 6.23). Again, jumps occur between \( 0 < t \leq 1 \) for the
Table 6.6: Moments of Network Completion Time With Case 2 Settings.

<table>
<thead>
<tr>
<th>Method</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>Run Time*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kulkarni</td>
<td>3.4378</td>
<td>14.6440</td>
<td>75.1273</td>
<td>1.7054</td>
</tr>
<tr>
<td>Lower P.R.</td>
<td>3.4346</td>
<td>14.6597</td>
<td>75.2103</td>
<td>0.0543</td>
</tr>
<tr>
<td>Upper P.R.</td>
<td>3.4524</td>
<td>14.5393</td>
<td>74.5621</td>
<td>0.0882</td>
</tr>
<tr>
<td>New L.B.</td>
<td>2.9653</td>
<td>-</td>
<td>-</td>
<td>0.0029</td>
</tr>
<tr>
<td>Devroye's L.B.</td>
<td>2.9167</td>
<td>-</td>
<td>-</td>
<td>0.0047</td>
</tr>
<tr>
<td>Devroye's U.B.</td>
<td>6.5247</td>
<td>48.6625</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Aven's U.B.1</td>
<td>5.2723</td>
<td>-</td>
<td>-</td>
<td>0.0070</td>
</tr>
<tr>
<td>Aven's U.B.2</td>
<td>5.3925</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

$M_k$ = the $k^{th}$ noncentral moment of network completion time ($k=1,2,3$).

* = CPU seconds on IBM 3081-D.
Table 6.7: Distribution Function of Network Completion Time in Test # 2

<table>
<thead>
<tr>
<th>Time</th>
<th>Kulkarni</th>
<th>Lower PR</th>
<th>Upper PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.00133</td>
<td>0.00075</td>
<td>0.02710</td>
</tr>
<tr>
<td>1.0</td>
<td>0.02149</td>
<td>0.01848</td>
<td>0.03466</td>
</tr>
<tr>
<td>1.5</td>
<td>0.08508</td>
<td>0.08423</td>
<td>0.07115</td>
</tr>
<tr>
<td>2.0</td>
<td>0.19715</td>
<td>0.19895</td>
<td>0.15297</td>
</tr>
<tr>
<td>2.5</td>
<td>0.32665</td>
<td>0.33691</td>
<td>0.27519</td>
</tr>
<tr>
<td>3.0</td>
<td>0.46481</td>
<td>0.47269</td>
<td>0.41765</td>
</tr>
<tr>
<td>4.0</td>
<td>0.69347</td>
<td>0.69182</td>
<td>0.68129</td>
</tr>
<tr>
<td>5.0</td>
<td>0.84062</td>
<td>0.83366</td>
<td>0.85332</td>
</tr>
<tr>
<td>6.0</td>
<td>0.92202</td>
<td>0.91669</td>
<td>0.94023</td>
</tr>
<tr>
<td>7.0</td>
<td>0.96317</td>
<td>0.96117</td>
<td>0.97726</td>
</tr>
<tr>
<td>8.0</td>
<td>0.98308</td>
<td>0.98304</td>
<td>0.99129</td>
</tr>
<tr>
<td>9.0</td>
<td>0.99234</td>
<td>0.99299</td>
<td>0.99619</td>
</tr>
<tr>
<td>10.0</td>
<td>0.99656</td>
<td>0.99724</td>
<td>0.99787</td>
</tr>
<tr>
<td>12.0</td>
<td>0.99932</td>
<td>0.99962</td>
<td>0.99877</td>
</tr>
<tr>
<td>15.0</td>
<td>0.99994</td>
<td>0.99998</td>
<td>0.99925</td>
</tr>
</tbody>
</table>
Figure 6.22 DF of Network Completion Time Obtained from Kulkarni and Adlakha's Backward Algorithm in Test #2.
Figure 6.23 DF of Network Completion Time Approximated by the Lower and Upper pr's in Test #2.
upper pr, due to the non-negligible value of the probability ($p_1=0.0269$) assigned to the first (degenerate) branch (see Figure 6.23). The difference in the shape of the density functions approximated by the lower and upper pr's can also be noticed in Figure 6.24. In general, the lower pr gives a better approximation than the upper pr in this three-level BAN with exponentially distributed processing times in which coefficients of variation are less than 1.

6.3.3 Case 3: Normally Distributed Processing Times

Recursion models (e.g., see Chapter 3) have been based on the assumption that processing times in production/assembly systems are normally distributed. It is interesting to compare the means and variances of operation finishing times approximated by recursion models with those approximated by the lower and upper pr's which were derived in this dissertation.

Test #3. The third test was designed to evaluate the accuracy of Model 2 in an eight-level BAN with independent and normally distributed node processing times.

Scenarios, defined by the means and coefficients of variations (cv's) of node processing times, were used:

(a) Balanced network with mean processing times at all I=255 nodes equal to 5.0 and cv equal to 0.25.
Figure 6.24 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Test #2.
(b) Imbalanced network with a (different) mean processing time at each node, which is randomly generated from a uniform distribution between 1.0 and 10.0, and cv=0.25 at all nodes.

With the assumption of normally distributed processing time at each node, the means and variances of operation finishing times can easily be approximated by recursion models which imbed the method developed by Clark (1961). Since the normal distribution is symmetric, the coefficient of skewness is equal to zero. Given the mean \(m\) and coefficient of variation \(c\) of the processing time, the first three noncentral moments \(m_1, m_2, m_3\) of the distribution of processing time may be easily shown to be

\[
\begin{align*}
  m_1 &= m, \\
  m_2 &= (1 + c^2)m_1^2, \\
  m_3 &= 3m_1m_2 - 2m_1^3.
\end{align*}
\]

The lower \(p_r\) may be used to represent the distribution of node processing time with known first three moments \(m_1, m_2, m_3\) as a mixture of two Erlang distributions of common order, and the distribution function of operation finishing times of nodes can then be approximated by Model 2. Approximated values of the means and variances of operation finishing times for balanced and imbalanced cases are listed in Table 6.8 and 6.9, respectively.

For both balanced and imbalanced cases, all three approximation methods give comparable approximated means and variances of operation finishing times. For instance, the absolute value of \(RD_1\) between the
Table 6.8: The Means and Variances of Operation Finishing Times of Nodes in an Eight-Level, Balanced BAN With Case 3 Settings.

<table>
<thead>
<tr>
<th>Level</th>
<th>Recursion Mean</th>
<th>Lower PR Mean</th>
<th>Upper PR Mean</th>
<th>New L.B.</th>
<th>Devroye’s U.B.</th>
<th>Aven’s U.B.</th>
<th>Devroye’s Upper bound on Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47.3062</td>
<td>4.6798</td>
<td>47.5068</td>
<td>5.0177</td>
<td>44.8504</td>
<td>40.0000</td>
<td>100.1484</td>
</tr>
<tr>
<td>2</td>
<td>41.0998</td>
<td>4.5729</td>
<td>41.2432</td>
<td>5.0177</td>
<td>39.1575</td>
<td>35.0000</td>
<td>75.2267</td>
</tr>
<tr>
<td>3</td>
<td>34.9141</td>
<td>4.4361</td>
<td>34.9998</td>
<td>4.8819</td>
<td>33.4645</td>
<td>30.0000</td>
<td>56.1956</td>
</tr>
<tr>
<td>7</td>
<td>10.7052</td>
<td>2.6276</td>
<td>10.6929</td>
<td>2.6559</td>
<td>10.6929</td>
<td>10.0000</td>
<td>11.7878</td>
</tr>
</tbody>
</table>

Run Time*  0.0142  727.4343  915.6443  8.1294  0.0390  0.0651

* = CPU seconds on IBM 3081-D.
Table 6.8 Continued:

<table>
<thead>
<tr>
<th>Level</th>
<th>Mean</th>
<th>Var.</th>
<th>L.B.</th>
<th>U.B.</th>
<th>AHW%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47.4124</td>
<td>5.1550</td>
<td>47.3832</td>
<td>47.4417</td>
<td>0.062</td>
</tr>
<tr>
<td>2</td>
<td>41.1468</td>
<td>4.9134</td>
<td>41.1182</td>
<td>41.1753</td>
<td>0.069</td>
</tr>
<tr>
<td>3</td>
<td>34.9440</td>
<td>4.6351</td>
<td>34.9163</td>
<td>34.9717</td>
<td>0.079</td>
</tr>
<tr>
<td>4</td>
<td>28.7715</td>
<td>4.3420</td>
<td>28.7447</td>
<td>28.7983</td>
<td>0.093</td>
</tr>
<tr>
<td>5</td>
<td>22.6599</td>
<td>3.9184</td>
<td>22.6344</td>
<td>22.6854</td>
<td>0.113</td>
</tr>
<tr>
<td>6</td>
<td>16.6125</td>
<td>3.3516</td>
<td>16.5889</td>
<td>16.6361</td>
<td>0.142</td>
</tr>
<tr>
<td>7</td>
<td>10.7119</td>
<td>2.6111</td>
<td>10.6910</td>
<td>10.7327</td>
<td>0.194</td>
</tr>
</tbody>
</table>

AHW% = 100 (half-width of the 99.5% confidence interval/mean)

+ = # of replications = 40,000

* = CPU seconds on IBM3081-D = 598.40
### Table 6.9: The Means and Variances of Operation Finishing Times of Nodes in an Eight-Level, Imbalanced BAN with Case 3 Settings

<table>
<thead>
<tr>
<th>Node</th>
<th>Recursion Mean</th>
<th>Lower PR Mean</th>
<th>Upper PR Mean</th>
<th>New L.B.</th>
<th>Devroye's Bounds on Mean</th>
<th>Aven's U.B.</th>
<th>Devroye's Upper bound on Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>67.7407</td>
<td>21.5874</td>
<td>67.8144</td>
<td>67.7786</td>
<td>62.856</td>
<td>64.260</td>
<td>145.4093</td>
</tr>
<tr>
<td>2</td>
<td>49.8508</td>
<td>7.8768</td>
<td>49.9863</td>
<td>49.9270</td>
<td>47.1486</td>
<td>45.3904</td>
<td>98.6662</td>
</tr>
<tr>
<td>3</td>
<td>57.9330</td>
<td>17.1164</td>
<td>57.9843</td>
<td>57.9563</td>
<td>55.5896</td>
<td>54.5901</td>
<td>110.5906</td>
</tr>
<tr>
<td>4</td>
<td>44.6239</td>
<td>9.6853</td>
<td>44.6859</td>
<td>44.6455</td>
<td>42.9395</td>
<td>42.0440</td>
<td>77.6743</td>
</tr>
<tr>
<td>5</td>
<td>44.7267</td>
<td>11.3251</td>
<td>44.7645</td>
<td>44.7309</td>
<td>42.5745</td>
<td>42.0346</td>
<td>77.7547</td>
</tr>
<tr>
<td>6</td>
<td>42.5657</td>
<td>11.0355</td>
<td>42.5794</td>
<td>42.5275</td>
<td>40.9085</td>
<td>40.5091</td>
<td>74.9402</td>
</tr>
<tr>
<td>7</td>
<td>49.8648</td>
<td>14.7866</td>
<td>49.8902</td>
<td>49.8682</td>
<td>47.6880</td>
<td>46.8937</td>
<td>84.4937</td>
</tr>
<tr>
<td>8</td>
<td>36.6724</td>
<td>8.8464</td>
<td>36.6981</td>
<td>36.6629</td>
<td>35.3591</td>
<td>34.2716</td>
<td>58.6169</td>
</tr>
<tr>
<td>9</td>
<td>36.6199</td>
<td>12.7430</td>
<td>36.6268</td>
<td>36.6608</td>
<td>36.0337</td>
<td>35.9199</td>
<td>56.1834</td>
</tr>
<tr>
<td>10</td>
<td>35.6224</td>
<td>8.3767</td>
<td>35.6225</td>
<td>35.5922</td>
<td>33.9602</td>
<td>33.4312</td>
<td>57.1734</td>
</tr>
</tbody>
</table>

**Run Time**: 0.0152 378.0454 507.6526 53.4835 0.0381 0.0649

* = CPU seconds on IBM 3081-D
### Table 6.9 Continued

<table>
<thead>
<tr>
<th>Node</th>
<th>Mean</th>
<th>Var.</th>
<th>Simulation**</th>
<th>L.B.</th>
<th>U.B.</th>
<th>AHW%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>67.8124</td>
<td>22.6107</td>
<td>67.7511</td>
<td>67.8736</td>
<td>0.090</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>49.9682</td>
<td>9.1720</td>
<td>49.9292</td>
<td>50.0072</td>
<td>0.078</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>57.9791</td>
<td>18.1419</td>
<td>57.9243</td>
<td>58.0340</td>
<td>0.095</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>44.6890</td>
<td>10.7104</td>
<td>44.6469</td>
<td>44.7312</td>
<td>0.094</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>44.7527</td>
<td>12.3376</td>
<td>44.7074</td>
<td>44.7979</td>
<td>0.101</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>42.5827</td>
<td>11.7680</td>
<td>42.5385</td>
<td>42.6269</td>
<td>0.104</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>49.9056</td>
<td>15.6587</td>
<td>49.8546</td>
<td>49.9565</td>
<td>0.102</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>36.6899</td>
<td>9.6601</td>
<td>36.6499</td>
<td>36.7299</td>
<td>0.109</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>36.6189</td>
<td>13.0852</td>
<td>36.5723</td>
<td>36.6655</td>
<td>0.127</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>35.6380</td>
<td>9.1778</td>
<td>35.5989</td>
<td>35.6770</td>
<td>0.109</td>
<td></td>
</tr>
</tbody>
</table>

AHW% = 100 (half-width of the 99.5% confidence interval/mean)

+ = # of replications = 40,000

* = CPU seconds on IBM 3081-D = 597.68
recursion model and the lower pr is 0.4285% (0.1087%) for the balanced (imbalanced) case.

In comparison with simulation estimates, approximated mean operation finishing times for nodes in the first three levels obtained from the recursion model and both pr's did not fall within the 99.5% confidence interval for the balanced case (see Table 6.8). It is to be noted that the AHW% of the confidence intervals are less than 0.08. For the imbalanced case, all (six) approximated mean operation finishing times of the first ten nodes (see Table 6.9) obtained from the lower (upper) pr fall within the 99.5% confidence intervals, each with a relatively small AHW% (the largest AHW% is less than 0.13). For both balanced and imbalanced cases, the approximated variances of operation finishing times obtained from both pr's are relatively close to the simulation estimated variances with the lower pr giving the closer of the two approximations, this improves the disadvantage of the recursion model which cannot approximate the variances very well.

For both balanced and imbalanced scenarios, the approximating density functions of network completion time obtained from the lower and upper pr's are almost identical and the shapes of these density functions are similar to normal distributions with relatively small variances (see Figures 6.25 and 6.26).

The CPU time for the recursion model is only 0.0142 (0.0152) seconds whereas the runtimes for the lower and upper pr's are 727.4343 (378.0454) and 915.6443 (507.6526) seconds, respectively, for the balanced (imbalanced) case. The large amount of runtime required by the lower and
Figure 6.25  PDF of Network Completion Time Approximated by the Lower and Upper pr's in Test #3 (Balanced Scenario)
Figure 6.26 PDF of Network Completion Time Approximated by the Lower and Upper pr's in Test #3 (Imbalanced Scenario)
upper pr's is due to the extremely high orders of the mixtures of Erlang distributions, which approximate operation finishing times of nodes in the higher levels (e.g., level one) obtained from the lower and upper pr's. For instance, the order number of the approximating df of network completion time obtained from the lower pr is 440 (205) for the balanced (imbalanced) case. It is suggested that the recursion model is preferred if we are primarily interested in the approximating means of operation finishing times of nodes in a BAN with independent and normally distributed node processing times.

6.4 Tightness of Model 3

In this section, the tightness of Model 3 bounds on the expected values of operation starting and finishing times is evaluated relative to different types of processing time distributions (e.g., mixtures of Erlangs, exponential and normal distributions).

6.4.1 Case 1: Processing Times are Distributed as Mixtures of Erlangs

Test # 4: This test evaluated the tightness of Model 3 bounds in the four-level and eight-level, balanced BAN's with the same parameter settings used in Test #1(a) and Test #1(b), respectively.

For both four-level and eight-level, balanced BAN's with Case 1 settings, the lower bounds on the expected values of operation finishing times of nodes in each level obtained from the new method developed in
this dissertation are always tighter than those from Devroye's method (see Tables 6.2 and 6.4). Since bounds become "loose" as the number of levels in the BAN increases, the RD\textsubscript{1} between the lower bounds and approximations will increase as the number of levels increases. For instance, the maximum RD\textsubscript{1} between new (Devroye's) lower bound and the lower pr are 17.035% (44.335%) and 28.536% (54.544%) for the four-level and eight-level BAN, respectively.

As expected, upper bounds on the expected values of operation finishing times obtained from Aven's two formulas are the same for both four-level and eight-level balanced BAN's in which the processing times of all nodes are independent and identically distributed. Tables 6.2 and 6.4 also show that upper bounds on the mean operation finishing times obtained from Aven's methods are always tighter than those derived from Devroye's method, but these upper bounds become "loose" rapidly as the number of levels increases. These large errors arise because Devroye's (1979) and Aven's (1985) methods are distribution-free; they are based simply on the means and variances of node processing times. The same result occurs for Devroye's upper bound on the second moment of operation finishing times.

For the four-level (eight-level), balanced BAN, the average runtime over the four runs for the new lower bound is 0.0262 (0.5526) seconds, the average runtimes for Devroye's and Aven's bounds are 0.0124 (0.0406) and 0.0182 (0.0664) seconds, respectively.
6.4.2 Case 2: Exponentially Distributed Processing Times

Test # 5 : This test evaluated the tightness of Model 3 bounds in a three-level, imbalanced BAN with the same parameter settings used in Test #2.

For the three-level, imbalanced BAN with Case 2 settings, the lower bound on the expected value of network completion time obtained from the new method developed in this dissertation is tighter than that derived from Devroye's method (see Table 6.6). The upper bound on the mean network completion time obtained from Aven's first method is tighter than that from his second method and that from Devroye's method. The CPU times for the new lower bound, Devroye's and Aven's bounds are very small and may be found in Table 6.6.

6.4.3 Case 3: Normally Distributed Processing Times

Test # 6 : This test evaluated the tightness of Model 3 bounds in an Eight-level BAN with the same parameter settings used in Test #3.

For both balanced and imbalanced BAN's with Case 3 settings, the lower bounds on the expected values of operation finishing times obtained from the new method developed in this dissertation are very close to those approximated by the recursion model and the lower and upper pr's discussed in section 6.3.2. Again, this new lower bound is better than Devroye's lower bound (see Table 6.8 and 6.9). The upper bounds on
the expected values of operation finishing times obtained from Aven's two methods are much better than those from Devroye's method. As expected, the methods of Devroye and Aven perform much better in this case with normally distributed processing times; it is conjectured that this performance results from the fact that the normal distribution is determined by just its mean and variance. Upper bounds on the variances of operation finishing times become weaker rapidly as the number of levels in the BAN increases.

For the balanced (imbalanced) BAN, the CPU time for the new lower bound is 8.1294 (53.4835) seconds; runtimes for Devroye's and Aven's bounds are 0.0390 (0.0381) and 0.0651 (0.0649) seconds, respectively. Since the df of node processing time, a normal distribution, is approximated by a mixture of two Erlang distributions of common order r and r is large (e.g., r ≥ 20), the time required to calculate the new lower bounds on the means of operation starting times increases rapidly [see equation (5.25) in Theorem 4] and is much greater than that required by Devroye's and Aven's bounds. It is interesting to notice that the runtime for the recursion model is less than that for the methods of Devroye and Aven.

6.5 Summary

The shapes of approximating density functions obtained from the lower and upper pr's are evaluated in several tests, all of which show that a unimodal (bimodal) approximating density function, in which the
coefficient of variation is less than or equal to 1, may be obtained from the upper $p_r$ by setting the lower limit $A$ to be zero (a positive real number). The Weibull density function cannot be well approximated by either of the $p_r$'s. The upper $p_r$ approximates the gamma density function very well, but the lower $p_r$ gives good approximations only if the coefficient of variation is less than or equal to 1.

The accuracy of Model 2 and tightness of Model 3 of the BAN are evaluated relative to different types of processing time distributions (e.g., mixtures of Erlangs, exponential and normal). All tests indicate that Model 2 gives good approximations of the distributions of operation finishing times and Model 3 gives tighter lower bounds on the expected values of operation starting and finishing times compared with those obtained from existing methods.
CHAPTER VII

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary

This dissertation develops a recursion procedure to model material flow in a cellular, small-lot assembly system and models to describe time dependent operations (i.e., material flow) in a system in which materials are assembled according to relationships indicated by a binary assembly network (BAN). The recursion model is an event-based model which parallels certain concepts of discrete-event simulation and ranks events according to the expected time at which they will occur. The recursion procedure is based on the insight that an activity (i.e., robot operation) can begin when all required resources (e.g., robot, lot, machine) are ready. That is, the activity beginning time is determined by the maximum of those resource ready times.

It is assumed that machine processing and robot service times are mutually independent, normally distributed random variables. The joint distribution of the activity beginning times and resource ready times is approximated by the multivariate normal. Clark's (1961) equations are then used to approximate the moments of the beginning time of each (robot) activity. To apply Clark's equations, pairwise correlations among resource ready times, which may be estimated from the correlations between certain activity beginning times and associated beta factors, must
be estimated. In order to efficiently estimate required correlations, a "fill-in on the fly" procedure, which works backwards over the completed sequence of activities and computes only the necessary correlations, is developed.

Accuracy of the recursion model was evaluated relative to a number of system features (e.g., lot routing, lot sequencing, parallel machines and finite buffer capacities). The recursion model gives good estimates of performance measures (e.g., makespan) compared with those derived from simulation. For example, the average (MD%) and mean square (MSD%) percent differences between recursion and simulation estimates of the mean values of all activity beginning times are relatively small in all tests. The recursion model has also been applied to a hypothetical industrial setting; small MD% and MSD% values resulted in all tests. Figures also show that the relative difference between recursion and simulation estimates of expected activity beginning time converges to a small limiting value as the number of activities increases.

For an L-level BAN in which processing times are distributed as mixtures of Erlangs, the moments of operation starting time of node i, $S_i$, in the $(L-m)_{th}$ $(m=1,2,...,L-1)$ level may be obtained by simply evaluating the maximum of two independent random variables, $F_{2i}$ and $F_{2i+1}$. Model 2 in this dissertation applies the properties of Tchebycheff systems and derives the lower and upper principal representations (pr's) to approximate the distributions of operation finishing times, $F_{2i}$ and $F_{2i+1}$, as mixtures of Erlangs. Hence, operation starting, $S_i$ [$S_i=\max(F_{2i},F_{2i+1})$], and finishing, $F_i$ ($F_i=S_i+P_i$), times of node i may be modeled.
The shapes of approximating density functions obtained from the lower and upper pr's are evaluated in several tests, all of which indicate that a unimodal (bimodal) approximating density function, in which the coefficient of variation is less than or equal to 1, may be obtained from the upper pr by setting the lower limit A to be zero (a positive real number). The Weibull density function cannot be well approximated by either of the pr's. The upper pr approximates the gamma density function very well, but the lower pr gives good approximations only if the coefficient of variation is less than or equal to 1.

The accuracy of Model 2 was also evaluated relative to a BAN with different types of processing time distributions. Model 2 gave good approximations (obtained from both lower and upper pr's) of the distributions of operation finishing times as noted by comparing with simulation model in Case 1 (processing times are distributed as mixtures of Erlangs), with exact results from Kulkarni and Adlakha's method in Case 2 (exponentially distributed processing times) and with the recursion and simulation models in Case 3 (normally distributed processing times). From the results of numerical evaluation of Model 2, it is conjectured that the lower (upper) pr gives a better approximation of the distribution of any nonnegative random variable with the coefficient of variation \( \leq (>) 1 \).

Based on the fundamental relationship between operation starting time of node \( i, S_i, (i=1,2,\ldots,2^{(L-1)}-1) \) and finishing times, \( F_{2i} \) and \( F_{2i+1} \), [i.e., \( S_i=\max(F_{2i},F_{2i+1}) \)] in an \( L \)-level BAN in which processing times are distributed as mixtures of Erlangs, new lower bounds (i.e., Model 3) on the expected values of operation starting, \( E(S_i) \), and finishing, \( E(F_i) \), times are
derived. The tightness of Model 3 bounds is also evaluated in the three test cases used to evaluate Model 2 accuracy. In all test cases, Model 3 gives tighter lower bounds on the expected values of operation starting and finishing times than those derived from existing methods (i.e., those of Devroye and Aven).

7.2 Conclusions

The analysis of transient performance is of much practical importance due to the fact that assembly systems rarely operate under steady state conditions. The recursion model (Model 1) developed in this dissertation extends the capabilities of earlier recursion approaches to model the transient performance (i.e., time dependent material flow) in cellular, small-lot assembly systems with a broad variety of features, including job-shop routing, finite input/output buffers, lot/machine sequencing, material handling and parallel machines.

The accuracy of Model 1 was evaluated relative to a number of system features, including lot routing, parallel machines, the degree of balance among processing times, lot sequencing, and finite input/output buffers. In all cases, Model 1 gave good estimates of system performance compare with those derived from simulation. Application of the recursion model in a hypothetical industrial setting demonstrates the applicability of this approach in practical production/assembly systems. The recursion model can be used to evaluate both alternative designs and operating policies in a cellular, small-lot assembly system. In particular,
the sequence of activities generated by the recursion model may be used as an aid in determining a short-term schedule.

Model 2 was developed to approximate the distributions of operation finishing times in a BAN; central to Model 2 are the lower and upper pr's. Each of the two pr's may be used to approximate the distribution of a random variable with support on \([0, \infty)\). It is conjectured that the lower (upper) pr gives better fits to unimodal distributions (e.g., Weibull and gamma) with the coefficient of variation \(\leq (>) 1\). Tests evaluating Model 2 relative to different types of processing time distributions indicate that Model 2 gives good approximations. In all test cases, new lower bounds (from Model 3) on the expected values of operation starting and finishing times were tighter than those available from prior methods.

Both Models 2 and 3 can be used to evaluate time dependent operations in an assembly system, assisting the material flow manager in planning and controlling an effective assembly schedule and in coordinating material flows to meet the scheduled performance in a stochastic assembly environment. In addition, Models 2 and 3 can also be used to evaluate an appropriate control tactic to achieve or to reschedule the due date of a product. These approximation methods and bounds could be widely applied, since the distribution of any random variable with support on \([0, \infty)\) may, in general, be arbitrarily closely approximated by a mixture of Erlang distributions.
7.3 **Recommendations for Future Research**

Models 2 and 3 developed in this research could be used to describe time dependent operations in a BAN. An interesting and important research problem is to extend these models to model transient performance (i.e., time dependent material flow) in a small-lot BAN, in which lots follow one another sequentially at stations in the assembly facility, when the processing (or operation) times are distributed as mixtures of Erlangs.
Proof of Lemma 1: Derivation of the $n^{th}$ moment of $Y_{jk} = \max[X_{1j}, X_{2k}]$, in which $X_{ij}$ is an $r_{ij}$th order Erlang distribution with rate $\mu_{ij}$.

Suppose the distribution functions of $X_{1j}$ and $X_{2k}$ are $G_1$ and $G_2$, respectively, and the density functions of $X_{1j}$ and $X_{2k}$ are $g_1$ and $g_2$, respectively. Then, the $n^{th}$ noncentral moment of $Y_{ij}$, $E(Y_{ij}^n)$, may be obtained as follows:

$$E(Y_{jk}^n) = \int_0^\infty t^n \Pr(Y_{jk} \leq t) dt \quad (A.1)$$

in which,

$$g_{ij}(t) = \frac{\mu_{ij}^{r_{ij}}}{(r_{ij} - 1)!} \cdot t^{r_{ij} - 1} \cdot e^{-\mu_{ij} t}$$

and

$$G_{ij}(t) = 1 - e^{-\mu_{ij} t} \sum_{m=0}^{r_{ij} - 1} \frac{(\mu_{ij} t)^m}{m!}$$

Now, consider the first term of right hand side of equation (A.1); the second term may be derived in a similar way.

$$\int_0^\infty t^n g_{ij}(t)G_{ij}(t) dt = \int_0^\infty t^n g_{ij}(t) dt - \frac{\mu_{ij}^{r_{ij}}}{(r_{ij} - 1)!} \sum_{m=0}^{r_{ij} - 1} \frac{\mu_{2k}^m}{m!} \int_0^\infty t^{n + m + r_{ij} - 1} \cdot e^{-(\mu_{ij} + \mu_{nj}) t} dt$$
\[
= \int_0^r t^n g_1(t) dt - \frac{\mu_{ij}^r}{(r_{ij} - 1)!} \sum_{m=0}^{r_{ij} - 1} \frac{\mu_{2k}^m (n + m + r_{ij} - 1)!}{m! (\mu_{ij} + \mu_{2k})^{n+m+r_{ij}}} \\
= E(X_{1,i}^n) - \eta_n(1, j, k)
\]  
(A.2)

Equation (5.8) and (5.9) in Lemma 1 may be obtained immediately from equation (A.2).

[2] **Proof of Proposition 1:** Derivation of the lower principal representation of a nonnegative rv, \( T \), with support on \([0, \infty)\) by matching its first three moments \((m_1, m_2, m_3)\):

The first three moments of the lower pr are:

\[
m_1 = p_1 \cdot t_1 + (1 - p_1) \cdot t_2 \quad \text{(A.3)}
\]
\[
m_2 = p_1 \cdot t_1^2 + (1 - p_1) \cdot t_2^2 \quad \text{(A.4)}
\]
\[
m_3 = p_1 \cdot t_1^3 + (1 - p_1) \cdot t_2^3 \quad \text{(A.5)}
\]

The relationships among \(m_1, m_2, m_3\) indicated by equations (A.3) - (A.5) may be represented as follows:

\[
m_2 - m_1^2 = p_1(1 - p_1)(t_1 - t_2)^2 \quad \text{(A.6)}
\]
\[
m_3 - m_1m_2 = p_1(1 - p_1)(t_1 - t_2)^2(t_1 + t_2) \quad \text{(A.7)}
\]
\[
m_2^2 - m_1m_3 = - p_1(1 - p_1)t_1t_2(t_1 - t_2)^2 \quad \text{(A.8)}
\]

Dividing equation (A.7) [(A.8)] by (A.6), equation (A.9) [(A.10)] may be obtained:

\[
\frac{m_3 - m_1m_2}{m_2 - m_1^2} = t_1 + t_2
\]
(A.9)
\[ \frac{m_2^2 - m_1 m_3}{m_2 - m_1^2} = -t_1 t_2 \]  \hfill (A.10)

For simplicity, let \( x = t_1 + t_2 \), and \( y = -t_1 t_2 \). \( t_1 \) and \( t_2 \) may be represented as:

\[ t_1 = \frac{x - \sqrt{x^2 + 4y}}{2}, \quad t_2 = \frac{x + \sqrt{x^2 + 4y}}{2} \]

and \( p_1 \) may be obtained from equation (A.3) as:

\[ p_1 = \frac{m_1 - t_2}{t_1 - t_2} = \frac{x + \sqrt{x^2 + 4y} - 2m_1}{2 \sqrt{x^2 + 4y}} \]

Parameters \( t_1, t_2 \) and \( p_1 \) in Proposition 1 follow immediately.

[3] **Proof of Proposition 2:** Derivation of the upper principal representation of a nonnegative rv, \( T \), with support on \([0,B]\) by matching its first three moments \((m_1, m_2, m_3)\):

The first three moments of the upper pr are:

\[ m_1 = p_u t_u + q_u B \] \hfill (A.11)
\[ m_2 = p_u t_u^2 + q_u B^2 \] \hfill (A.12)
\[ m_3 = p_u t_u^3 + q_u B^3 \] \hfill (A.13)

Rearranging equations (A.11) - (A.13) as follows:

\[ m_1 - q_u B = p_u t_u \] \hfill (A.14)
\[ m_2 - q_u B^2 = p_u t_u^2 \] \hfill (A.15)
\[ m_3 - q_u B^3 = p_u t_u^3 \] \hfill (A.16)

Dividing equation (A.14) by (A.15) and (A.15) by (A.16),
\[
\frac{1}{t_u} = \frac{m_1 - q_u B}{m_2 - q_u B^2} = \frac{m_2 - q_u B^2}{m_3 - q_u B^3},
\]

i.e., \((m_1 - q_u B)(m_3 - q_u B^3) = (m_2 - q_u B^2)^2\).

Hence, \(q_u\) may be obtained by solving the above equation. \(p_u\) and \(t_u\) may also be obtained immediately from equations (A.14) and (A.15).

[4] **Proof of Theorem 3:** Derivation of the upper principal representation of a nonnegative rv, \(T\), with support on \([A,B]\) by matching its first three moments \((m_1,m_2,m_3)\):

From equation (5.21), the first three moments of the upper pr are:

\[
\begin{align*}
    m''_1 &= p_u(t_u - A) + q_u B', \\
    m''_2 &= p_u(t_u^2 - A^2) + q_u B'_2, \\
    m''_3 &= p_u(t_u^3 - A^3) + q_u B'_3.
\end{align*}
\]

Rearranging equations (A.17) - (A.19) as follows:

\[
\begin{align*}
    m''_1 - q_u B' &= p_u(t_u - A), \\
    m''_2 - q_u B'_2 &= p_u(t_u^2 - A^2), \\
    m''_3 - q_u B'_3 &= p_u(t_u^3 - A^3).
\end{align*}
\]

For simplicity, let \(f_i(q_u) = m''_i - q_u B'_i\), \((i = 1,2,3)\). Dividing (A.21) by (A.20), \((t_u + A)\) may be represented as follow.

\[
t_u + A = \frac{f_2(q_u)}{f_1(q_u)}
\]

(A.23)
In order to construct a form similar to equation (A.23) using equations (A.21) and (A.22), let \( \omega = p_u t_u \), such that \( p_u = \omega / t_u \). From Equations (A.21) and (A.22), \( f_2(q_u) \) and \( f_3(q_u) \) may be represented as:

\[
f_2(q_u) = \omega t_u - \frac{\omega A^2}{t_u}
\]

\( (A.24) \)

\[
f_3(q_u) = \omega t_u^2 - \frac{\omega A^3}{t_u}
\]

\( (A.25) \)

From equation (A.20), \( f_1(q_u) \) may be represented as \( f_1(q_u) = \omega - \frac{\omega A}{t_u} \), such that

\[
\frac{\omega A}{t_u} = \omega - f_1(q_u)
\]

\( (A.26) \)

Substituting (A.26) into (A.24) and (A.25), the following two equations may be obtained:

\[
f_2(q_u) - A f_1(q_u) = \omega(t_u - A)
\]

\( (A.27) \)

\[
f_3(q_u) - A^2 f_1(q_u) = \omega(t_u^2 - A^2)
\]

\( (A.28) \)

Dividing (A.28) by (A.27), \( t_u + A \) may be represented as follows.

\[
t_u + A = \frac{f_3(q_u) - A^2 f_1(q_u)}{f_2(q_u) - A f_1(q_u)}
\]

\( (A.29) \)

From equations (A.23) and (A.29),

\[
\frac{f_2(q_u)}{f_1(q_u)} = \frac{f_3(q_u) - A^2 f_1(q_u)}{f_2(q_u) - A f_1(q_u)}
\]

Hence, \( f_2(q_u) [f_2(q_u) - A f_1(q_u)] = f_1(q_u) [f_3(q_u) - A^2 f_1(q_u)] \), i.e.,
\[ Xq_u^2 - Yq_u + Z = 0 \]  \hspace{1cm} (A.30)

in which,

\[ X = B'_1 B'_3 - B'_2^2 + A B'_1 B'_2 - A^2 B'_1 \]
\[ Y = m''_1 B'_3 - 2m''_2 B'_2 + m''_3 B'_1 + A m''_2 B'_1 + A m''_1 B'_2 - 2A^2 m''_1 B'_1 \]
\[ Z = m''_1 m''_3 - m''_2^2 + A m''_1 m''_2 - A^2 m''_1 \]

Solving equation (A.30), \( q_u \) may be obtained. \( t_u \) and \( p_u \) in Theorem 3 may be obtained immediately.

[5] Proof of Theorem 4: Derivation of \( E(\max [X_1, X_2 + w_2 - w_1]) \)

The derivation of \( E(\max [X_1, X_2 + w_2 - w_1]) \) is based on the following four results.

Result 1:  \[ \int_{a}^{\infty} (t - a)^{\beta} e^{-\lambda t} dt = e^{-\lambda a} \frac{\beta!}{\lambda^{\beta+1}} \]

Proof: \[ \int_{a}^{\infty} (t - a)^{\beta} e^{-\lambda t} dt = e^{-\lambda a} \int_{a}^{\infty} (t - a)^{\beta} e^{-\lambda(t - a)} dt \]
let \( t - a = u \),
\[ = e^{-\lambda a} \int_{0}^{\infty} u^{\beta} e^{-\lambda u} du . \]

Hence, result 1 is obtained.

Result 2:  \[ \int_{a}^{\infty} t^{\alpha} e^{-\lambda t} dt = \frac{\alpha!}{\lambda^{\alpha+1}} e^{-\lambda a} \sum_{n' = 0}^{\alpha} \frac{(\lambda a)^{n'}}{n'!} \]

Proof:
\[ \int_{a}^{b} e^{-\lambda t} dt = \int_{0}^{b} e^{-\lambda t} dt - \int_{0}^{a} e^{-\lambda t} dt \]  
(A.31)

the first term in equation (A.31) is equal to \( \alpha! / \lambda^{\alpha+1} \).

Integrating the second term by parts gives

\[ \int_{0}^{b} e^{-\lambda t} dt = \frac{\alpha!}{\lambda^{\alpha+1}} - \frac{\alpha!}{\lambda^{\alpha+1}} \sum_{n=0}^{a} \frac{(\lambda a)^{n'}}{n'} \]

hence, result 2 is obtained.

Result 3:

\[ \int_{a}^{b} (t - a)^{\beta} e^{-\lambda t} dt \]

\[ = \frac{\alpha! \beta!}{\lambda^{\alpha+\beta+1}} e^{-\lambda a} \left[ \sum_{n=1}^{a} \frac{(\alpha + \beta - 1 - n)!}{(\alpha - n)! (\beta - 1)!} \frac{(\lambda a)^{n'}}{n'} + \frac{(\alpha + \beta - 1)!}{\alpha!(\beta - 1)!} \right] \]

Proof:

By using results 1 and 2, integrating \( \int_{a}^{b} (t - a)^{\beta} e^{-\lambda t} dt \) by parts, result 3 may be obtained.

Result 4:

\[ E \{ \max [X_1, X_2 + w_2 - w_1] \} \]

\[ = \int_{w_2 - w_1}^{w} t g_1(t) G_2(t - w_2 + w_1) dt + \int_{w_2 - w_1}^{w} t g_2(t - w_2 + w_1) G_1(t) dt \]

\[ = (u_1 - u_2) + (u_3 - u_4) \]  
(A.32)

a. \( \int_{w_2 - w_1}^{w} t g_1(t) G_2(t - w_2 + w_1) dt \)

\[ = \int_{w_2 - w_1}^{w} t g_1(t) dt - \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mu_{1j}^{r_1} P_{1j} P_{2k} (r_{1j} - 1)! e^{-\mu_{1k} (w_1 - w_2)} \cdot \]

\[ \sum_{m=0}^{r_{1k}} \frac{\mu_{2k}^{m}}{m!} \int_{w_2 - w_1}^{w} t^{r_{1j}} [t - (w_2 - w_1)]^{m} e^{-\mu_{1}^{w_2} + \mu_{1}^{w_1}} dt \]
Applying results 2 and 3, \( u_1 \) and \( u_2 \) in Theorem 4 may be obtained.

\[
b. \quad \int_{w_2 - w_1}^{\infty} t g_2(t - w_2 + w_1)G_1(t) \, dt
\]

\[
= \int_{w_2 - w_1}^{\infty} tg_2 \{t - (w_2 - w_1)\} \, dt - \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} P_{2k}P_{1j} \frac{\mu_{2k}^{r_{2k}}}{(r_{2k} - 1)!} e^{-\mu_{2k}(w_1 - w_2)} .
\]

\[
\sum_{m=0}^{r_{1j} - 1} \frac{\mu_{1j}^m}{m!} \int_{w_2 - w_1}^{\infty} t^{m+1} \{t - (w_2 - w_1)\}^{r_{2k} - 1} e^{-\mu_{2k} + \mu_{1j}} \, dt
\]

Applying results 1 and 3, \( u_3 \) and \( u_4 \) in Theorem 4 may be obtained.
REFERENCES


