Performance evaluation of computer controlled systems with shared and limited resources.

Diwakar Gupta
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/3051

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
CANADIAN THESES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

THÈSES CANADIENNES

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE
PERFORMANCE EVALUATION OF COMPUTER CONTROLLED SYSTEMS WITH
SHARED AND LIMITED RESOURCES

by

Diwakar Gupta

A thesis
presented to the University of Windsor
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in
Department of Industrial Engineering

Windsor, Ontario, 1984
(c) Diwakar Gupta, 1984
The University of Windsor requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.
ABSTRACT

Some exact and approximate analytic procedures for solving queueing networks often encountered while designing computer-controlled systems have been presented in this work. A thorough review of literature reveals that the available methods are not appropriate for solving such queueing networks as they are characterised by finite queue capacities and contention among customers for shared and limited resources. Computerised Manufacturing Systems (CMSs), among others, are examples of such queueing systems. A Semi-Markov modeling approach is then introduced. In this approach, the states are observed only at the times of transitions. The mean residence time in each state is calculated using the distribution function of the conditional state occupancy times. The state space corresponding to the exact model grows explosively as the system size increases resulting in lengthy calculations. Several appropriate states are lumped together in an effort to reduce the complexity of the original state description. The resulting models are approximate because lumpability conditions are not necessarily valid. Two levels of approximations are studied which are much simpler to analyse. For the second approximation, the transitions
between states are found to exhibit a pattern that can be exploited to derive efficient computational algorithms. Several performance measures are then evaluated using the steady state probability distribution of the Semi-Markov process. A benchmark example of the CMSs is analysed in detail and parametric analysis is carried out using the three analytic techniques and simulation. The analytical results lie within 99% confidence interval on results of simulation experiments. The approximate models 1 and 2 give very tight lower and upper bounds respectively on system throughput. Sensitivity analysis reveals that the approximations are very good for moderate to heavy load conditions. The approximate model solving techniques are very efficient and result in substantial savings in computer-time.

The present work is an important extension to the area of queueing systems analysis. The thesis introduces a new approach to the performance modeling of computer-controlled systems.
DEDICATION

To my parents.
ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deep gratitude towards Dr. M. Alam whose unfailing moral support and guidance through the ups and downs of this research has finally made it possible. I would also like to express my gratitude to Professor A.A. Danish for showing interest in my work, guidance and useful suggestions from time to time.

I wish to thank Dr. R.S. Lashkari and Dr. S.P. Dutta for inspiring me through the early phases of this work. A note of thanks to my committee members, Dr. R.S. Lashkari and Dr. P.N. Kaloni for reviewing my work.

I also wish to thank my friends, all the graduate students of the Department of Industrial Engineering who have been a constant help and have contributed through very useful discussions. A special note of thanks for Mr. & Mrs. E. Reimer whose affection helped me through emotional crises and whose faith in me was a constant source of inspiration to me. Finally, I wish to thank my friend, Miss Jyothi Seshadri for being an excellent companion and for providing encouragement, and moral courage.
TABLE OF CONTENTS

ABSTRACT ........................................................................ iv
DEDICATION ....................................................................... vi
ACKNOWLEDGEMENTS ...................................................... vii
LIST OF FIGURES ............................................................. x
LIST OF TABLES ............................................................... xii

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION .................................................. 1</td>
<td></td>
</tr>
<tr>
<td>General Introduction ............................................. 1</td>
<td></td>
</tr>
<tr>
<td>Definitions and Nomenclature: (Queueing Systems) ... 3</td>
<td></td>
</tr>
<tr>
<td>Definition of a CMS .............................................. 6</td>
<td></td>
</tr>
<tr>
<td>II. A REVIEW OF LITERATURE ................................. 9</td>
<td></td>
</tr>
<tr>
<td>Introduction ....................................................... 9</td>
<td></td>
</tr>
<tr>
<td>Previous Work Related to Queueing Networks .......... 9</td>
<td></td>
</tr>
<tr>
<td>Previous Work Related to Computerised Manufacturing Systems ........................................ 22</td>
<td></td>
</tr>
<tr>
<td>III. THE CMS MODEL .............................................. 31</td>
<td></td>
</tr>
<tr>
<td>Introduction ....................................................... 31</td>
<td></td>
</tr>
<tr>
<td>CMS Architecture .................................................. 32</td>
<td></td>
</tr>
<tr>
<td>A Queueing Network Model of the CMS ..................... 36</td>
<td></td>
</tr>
<tr>
<td>Available Methods of Solution .............................. 41</td>
<td></td>
</tr>
<tr>
<td>IV. EXACT AND APPROXIMATE MODELS USING THE SEMI-MARKOV APPROACH .................. 45</td>
<td></td>
</tr>
<tr>
<td>Introduction ....................................................... 45</td>
<td></td>
</tr>
<tr>
<td>The Semi-Markov Approach ................................... 47</td>
<td></td>
</tr>
<tr>
<td>Performance Measures ......................................... 52</td>
<td></td>
</tr>
<tr>
<td>The Exact Model .................................................... 53</td>
<td></td>
</tr>
<tr>
<td>The Approximate Models ....................................... 56</td>
<td></td>
</tr>
<tr>
<td>Model #1 ............................................................ 56</td>
<td></td>
</tr>
<tr>
<td>Model #2 ............................................................ 57</td>
<td></td>
</tr>
<tr>
<td>Examples &amp; Results ............................................... 58</td>
<td></td>
</tr>
<tr>
<td>Summary ............................................................. 82</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Local Balance Equation In Pictorial Form</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Representation of an E2 server by the method of Phases</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>A Non-Local Balance Network</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Schematic diagram of the central server model of a CMS showing item flow</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Queueing network model of the central station CMS with contention for AGVs</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>State transition diagram for a 5X2X1 CMS using the exact analytic model</td>
<td>68</td>
</tr>
<tr>
<td>4.2</td>
<td>State transition diagram for a 5X2X1 CMS using the approximate model #1</td>
<td>69</td>
</tr>
<tr>
<td>4.3</td>
<td>State transition diagram for a 5X2X1 CMS using the approximate model #2</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>Schematic of the SLAM network for a 5x2x1 CMS example</td>
<td>71</td>
</tr>
<tr>
<td>4.5</td>
<td>Asymptotic bounds on system delay using the exact model</td>
<td>74</td>
</tr>
<tr>
<td>4.6</td>
<td>Asymptotic bounds on system delay using the approximate model#1</td>
<td>75</td>
</tr>
<tr>
<td>4.7</td>
<td>Asymptotic bounds on system delay using the approximate model#2</td>
<td>76</td>
</tr>
<tr>
<td>4.8</td>
<td>99% Confidence Interval on results of simulation &amp; results of the exact model</td>
<td>78</td>
</tr>
<tr>
<td>4.9</td>
<td>99% Confidence Interval on results of simulation &amp; results of approximate model #1</td>
<td>79</td>
</tr>
<tr>
<td>4.10</td>
<td>99% Confidence Interval on results of simulation &amp; results of approximate model #2</td>
<td>80</td>
</tr>
</tbody>
</table>
4.11 System throughput as a factor of no. of jobs . . . . 81
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Nomenclature in Queueing Systems</td>
<td>5</td>
</tr>
<tr>
<td>4.1</td>
<td>System processing delay and throughput rate as a factor of number of jobs for an Nx2xl CMS with 2 AGVs using simulation, the exact model and the approximations 1 &amp; 2</td>
<td>73</td>
</tr>
</tbody>
</table>
Chapter I
INTRODUCTION

1.1 GENERAL INTRODUCTION

Computer-controlled systems have become an integral part of modern industrial complexes. These systems are often very large and complex and therefore cannot be analysed by intuition alone. Those charged with the task of designing such complex systems to meet the high standards of performance expected of them, must resort to abstract system conceptualisations and advanced mathematical tools in order to be able to predict their essential performance characteristics. Queueing networks models have been found to be very useful analytical tools for the system designers. They offer accuracy, simplicity of structure and ease of parameterisation.

The present work deals with some exact and approximate analytical models for queueing networks with blocking due to finite queue capacities and limited resources. It presents a Semi-Markov approach to analyse queueing network models associated with many computer systems such as the Computerised Manufacturing Systems
(CMSs). Throughout this thesis, emphasis has been laid on developing practical tools for analysis, and solution procedures that are computationally efficient, yet nearly exact. Applicability of these tools to the problems of CMSs is demonstrated through examples. The results of such an analysis could be used to evaluate performance characteristics as well as to develop a better understanding of the system.

Queueing network analysis has been used extensively in performance evaluation of computer systems; for example, see [20], [33] and [50]. Some examples of its use in the manufacturing industry include: [13], [14], [26], [43], [44], [54], and [63] etc. A major portion of these models form one class of queueing network models called the 'local balance' networks [11], [16], [19], [50], [24], and [30]. The methods available to analyse those networks that do not satisfy the requirements of local balance are very limited and complex. Many restrictive assumptions of some of the previous models have been relaxed in this work and efficient solution procedures have been presented. Computerised Manufacturing Systems (CMSs) are also known as Flexible Manufacturing Systems (FMSs) and the terms CMSs and FMSs are used interchangeably in this thesis. FMSs are a fairly recent concept and seem to offer a promising solution to many problems of the modern
manufacturing industry. Since these systems are essentially computer-controlled systems, queueing network analysis is found to be very apt for evaluation and design of such systems. Several of these systems are analysed in the later part of this thesis through examples.

1.2 DEFINITIONS AND NOMENCLATURE: (QUEUEING SYSTEMS)

As a shorthand in describing queueing processes, a notation has evolved, due for most part to Kendall [35]. The use of this notation is standard throughout the queueing literature. A queueing process is described by a series of symbols and slashes such as $A/B^X/Y/Z$ where the symbols stand for the following:

- $A =$ Arrival (or interarrival time) distribution.
- $B =$ Departure (or service-time) distribution.
- $X =$ # of parallel service channels in the system.
- $Y =$ Maximum number allowed in the system (in service and waiting).
- $Z =$ Queue discipline.

Some standard symbols used for these characteristics are shown in Table 1.1. In addition to these, a superscript is attached to the first symbol if bulk arrivals exist and to second symbol if bulk service is used. To illustrate the use of this notation, consider $M^{(b)}/G/c/N/FIFO$ This denotes exponential interarrival times with b customers per arrival, general service times, c parallel servers, first-come-first-
served service discipline, and the maximum number allowed in the system at any time is equal to \( N \). In many situations only the first three symbols are used. Current practice is to omit the waiting capacity if no restriction is imposed \((Y=\infty)\) and to omit the queue discipline if it is first in first out \((Z=\text{FIFO})\). Thus \( M/D/2 \) would be used to represent a queueing system with exponential input, deterministic service, two servers, no limit on system capacity, and first-come, first-served queue discipline.

The designation does not classify series and networks of queues, balking of customers, state dependent service times, heterogeneity of servers, and a number of other variations of the queueing systems. Therefore such characteristics have to be mentioned explicitly along with the queue notation.

A network of queues is said to be Markovian if its interarrival and service-time distributions are exponential. Conversely, a non-Markovian network is such that either interarrival or interservice time or both is/are not exponentially distributed. This classification is significant because the solution procedure of Markovian networks becomes very simple owing to the Markov property of the exponential distribution.
TABLE 1.1
Nomenclature in Queueing Systems

<table>
<thead>
<tr>
<th>SYMBOLS</th>
<th>CHARACTERISTICS</th>
<th>EXPLANATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>A,B</td>
<td>Poisson arrival or departure distributions (or equivalently exponential inter-arrival or service time distributions; M refers to the Markov property of the distribution).</td>
</tr>
<tr>
<td>GI</td>
<td>A</td>
<td>General independent distribution of arrivals (or interarrival times).</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>General distribution of departures (or service times).</td>
</tr>
<tr>
<td>D</td>
<td>A,B</td>
<td>Deterministic departure or service times.</td>
</tr>
<tr>
<td>E_k</td>
<td>A,B</td>
<td>Erlangian or gamma interarrival or service time distribution with k phases (k=1,2,...).</td>
</tr>
<tr>
<td>1,2,...,∞</td>
<td>X</td>
<td># of parallel servers.</td>
</tr>
<tr>
<td>1,2,...,∞</td>
<td>Y</td>
<td>Restriction on system capacity.</td>
</tr>
<tr>
<td>FIPO</td>
<td>Z</td>
<td>First in first out queue discipline.</td>
</tr>
<tr>
<td>LIPO</td>
<td>Z</td>
<td>Last in first out queue discipline.</td>
</tr>
<tr>
<td>SIRO</td>
<td>Z</td>
<td>Service in random order.</td>
</tr>
<tr>
<td>SPT</td>
<td>Z</td>
<td>Shortest Processing Time</td>
</tr>
<tr>
<td>PRI</td>
<td>Z</td>
<td>Priority.</td>
</tr>
<tr>
<td>GD</td>
<td>Z</td>
<td>General discipline.</td>
</tr>
</tbody>
</table>
1.3 DEFINITION OF A CMS

The term CMS may be used to describe some simple systems like a single or small number of machines serviced by robots as well as very complex systems like multiple machines serviced by automated and programmable materials handling equipment. Because of such diversity of CMSs encountered in the modern industry, there is no universally accepted definition of a CMS. We describe a CMS after Buzacott and Shanthikumar [14] as a system "

...of machines where production operations are performed, linked by materials handling system and all under central computer control." Several case studies and other descriptions of CMSs can be found in literature; for example, see [1], [2], [3], [4], [5], [6], [7], [8], [9], [17], [22], [23], [25], [29], [59] and [60]. Generally a CMS is defined by its attributes and is a term applied to a manufacturing industry where;

1. A large variety of components are manufactured in relatively small batches.

2. A small number of Direct Numerical Control (DNC) machines are required.

3. Components are moved into, within, or out of the system by means of an automated materials handling device.
4. All the operational aspects of the system are controlled by an online computer on a real time basis.

This definition points to a modular structure in CMSs [40]. It is easier to study a CMS with respect to these modules. The three major categories of these modules and their typical components are:

1. Machining Modules:
   i) Head changer
   ii) Machining centre
   iii) Multi-module
   iv) Vertical turning lathe
   v) N/C milling
   vi) N/C turning
   vii) Head Indexer
   viii) Boring bar changer

2. Materials Handling Modules:
   i) Roller conveyor
   ii) In/Out shuttle
   iii) Thru shuttle
   iv) Guided Vehicle
   v) Shuttle car
   vi) Towline

3. System Control Modules:
   i) Computerised Numerical Control (CNC)
ii) Materials Handling Control

iii) Supervisory computer

Flexibility appears to be a concept that can have several meanings. It can, for example, include automated tool changing capability at versatile machines, or the adaptability of versatile machining centres to possible future changes in the production mix. From a materials handling point of view, a CMS includes flexible routing—the ability to accept a different part at a given machine without delay, and to route a given part over a different path to utilise the next free machine. This also involves optimal decisions on buffer storages ahead of machining stations to increase machine utilisation. CMSs offer a trade off between the versatility of stand alone DNC machines and the efficiency of high volume dedicated transfer lines.

Computerised Manufacturing Systems are very complex manufacturing systems. They are significant due to their great potential in solving many problems of the modern manufacturing industry [25]. These have created many new openings for active research in the area of analytical modeling.
Chapter II
A REVIEW OF LITERATURE

2.1 INTRODUCTION

This chapter is divided into two major parts. Since we are interested in finding techniques for performance evaluation of computer controlled systems using queueing network analysis, the following section will be devoted to a brief review of previous works related to the problems of queueing networks. Next, we shall discuss some of the important existing models used for CMSs with reference to their functions and limitations, and prepare the backdrop for proposing a new model in the next chapter.

2.2 PREVIOUS WORK RELATED TO QUEUEING NETWORKS

The limited amount of literature on steady state solution of multi channel queueing networks deals primarily with the Markovian networks. In a study on the output process of a queueing system, Burke [12] proved that the equilibrium distribution of the number of service completions in an arbitrary time interval is the same as the input distribution, for any number of servers for a queue with Poisson input and exponential holding times. This led to an
easy analysis of Markovian queueing systems because tandem queues could be analysed stage by stage in so far as the separate delay and queue length distributions were concerned.

In a parallel study by Reich [48] followed by [49], the same result has been established using a Markov chain approach. Reich's paper also proves that this result does not hold for a $E_2/E_2/1$ queueing process. Note that an $E_2$ distribution is nothing but a two dimensional exponential distribution. This paper emphasises that an easy solution to the example by Burke is only due to the Markovian nature of the exponential distribution.

Hunt [28] gave a method of solution for a sequence of waiting lines with Poisson arrivals and exponential service times. Four particular cases of service facility have been treated, involving no storage space between stages, infinite storage space between stages, finite storage space between stages, and the case of unpaced belt production. The analysis uses the recursive nature of steady state difference equations to evaluate such characteristics as; the steady state probability vector, time in the queue etc. The maximum utilisation possible under the above four cases have been compared. As expected, the paper establishes that maximum possible utilisation is far less when storage
space between stages is reduced owing to the increased blocking.

The work of Jackson [30] gives a solution procedure for a relatively complex network of Markovian queues. The model considers a variable number of servers at each stage of service and a positive probability of arrivals according to a Poisson type time series. After being served at any service node, the customer moves to the next node. The network has been modeled as a vector valued Markov chain defined on a countably infinite state space. A common problem with all the models cited so far is that the analysis breaks down as soon as non-Markovian arrival or service processes are encountered.

An approach that decomposes the queueing network into subnetworks to be analysed in isolation has been presented by Disney and Cherry [21]. Here the output of one subnetwork is taken as input to another subnetwork. Subsequently, stochastic properties of the flow representing the output of the network as a whole can be determined by recomposing. The fragmenting of a complex network into simple segments and then reuniting the outputs of several segments is achieved by 'Decomposition' and 'Recomposition' switches respectively. A decomposition switch is a point within the network at which an event in a single stochastic
stream is directed by a stochastic mechanism (e.g. first available vacant channel, channel with smallest queue etc. etc.) to one of a finite number of subsequent streams. A recomposition switch is a point within the network at which two or more stochastic streams are merged via some stochastic mechanism to form a single stochastic stream. However, as Disney and Cherry have also pointed out, not all networks are decomposable and there is no result that proves the necessary and sufficient conditions for a network to be decomposable. Therefore, the applicability of this model is limited. Moreover, in many cases it is easier to solve using the simplistic structure of multi-dimensional Markov processes.

Chandy et. al. [19], have identified the most general class of decomposable networks. The property that makes them separable is known as the 'Product form and Local Balance in Queueing Networks'. A queue is said to satisfy local balance if for every pair of (feasible) states $S$ and $S'$:

$$\text{transition rate from } S' \text{ to } S = \text{transition rate from } S \text{ to } S'$$

(2.1)

A pictorial representation of local balance is shown in fig. 2.1. For a network of $m$ queues where the $m^{th}$ queue has a set of classes referred to as $\mathcal{C}(m)$, the network is said to
satisfy local balance if all queues of the network satisfy local balance in isolation. In order to obtain equilibrium state probabilities of local balance networks, each local balance queue is analysed separately. Let $S_m$ be any feasible state of queue $m$ ($m=1,2,\ldots,M$) in isolation, then a state $S$ is a feasible network state if and only if $S = (S_1, S_2, \ldots, S_M)$ and the sum of the queue population of each queue in isolation is equal to the total population of the queue network. Now the equilibrium state probability of the network state $S$ can be shown to be equal to:

$$p(S) = p(S_1)p(S_2) \ldots p(S_M)$$

(2.2)

where $p(S_i)$ is obtained from analysing queues in isolation.

As expected, this concept makes the analysis of network of queues very simple. However, its applicability is restricted to networks satisfying local balance. Also the local balance is a characteristic of some Markov processes and is not applicable to non-Markovian processes. In order to analyse queues with more general Erlangian service time distributions, they are represented as 'stages of Exponential service'. Most general distributions can be approximated with fairly good amount of accuracy by having the right choice of the number of phases. Fig. 2.2 shows a
Figure 2.1: The local balance equation in pictorial form.
(Source: Sauer and Chandy [50])
representation of Erlang service time with two phases. The
network shown in fig. 2.3 with exponential inter-arrival and
service time distributions does not satisfy local balance
and therefore cannot be analysed using this model. Note that
this can be solved conveniently using the approach of
Jackson [30].

Newell [42] has been able to obtain general
results of exact analysis of non-Poisson or non-stationary
queueing systems using the diffusion approximation. It is
possible to obtain answers to practical problems using this
approach but the technique is very tedious and difficult to
implement.

There has been only a limited amount of work done
even in the area of multi-server queueing systems with a
single stage of service and general service time
distributions. Jackson [31], made an effort to solve an
$E_m/E_2/2$ queue with homogeneous servers using the generating
function technique. But this study, fails to make use of the
important cyclic property of the phase transitions.
Consequently, the method of obtaining numerical results
becomes very complex and tedious.

In an attempt to simplify the brute force
calculations required to solve the general $E_m/E_k/r$ systems,
Figure 2.2: Representation of an $E_2$ server by the method of phases. (Source: Sauer and Chandy [50])
Figure 2.3: A non-local balance network.
(Source: Sauer and Chandy [50])
Yu [64] has exploited the special repetitive pattern of phase transitions and has been able to obtain a simplified exact solution procedure. This model can also account for the heterogeneity of servers. However, the analysis becomes intractable for network of queues with several stages of service. Also, in such a case it is very difficult to recognize the special repetitive pattern of phase transitions.

Simulation has been a widely used technique for many problems requiring greater detail than what can be handled by available analytical models. The easy availability and abundance of computers and a number of software packages (for example see [45], [46], [47], etc.) for various applications have contributed towards making simulation one of the most popular techniques of analysis in the present day.

Some ([18], [32], [36] & [39]) very efficient approximation techniques have appeared in recent past due to application of queueing theory to computer systems design. Chandy et. al. [18] have presented an approach analogous to the Norton's theorem in electrical circuit theory. A closed queueing network with M exponential service stations and N customers is considered. In order to study the behaviour of a subsystem as its parameters are varied, an equivalent
network is constructed in which all the queues except those in the subsystem are replaced by a single composite queue. The service rate for the composite queue is set equal to the rate at which customers pass through the subsystem. Now the service times of all servers in the subsystem are set equal to zero. This is equivalent of shorting in the electrical circuits. Then, according to the Norton's theorem, the behaviour of the subsystem in the equivalent network is the same as in the initial network. This approach is now referred to as the Norton's theorem for queueing networks and holds for the local balance networks. It is also shown that the theorem does not necessarily hold for networks in which local balance does not exist. This method makes analysis of queueing networks computationally efficient and is also valid for open networks satisfying local balance.

Jacobson and Lazowska [32] describe an approximate solution technique for queueing networks that involve simultaneous or overlapped possession of resources and name it the 'method of surrogates'. The instance of simultaneous resource possession occurs when a customer obtains a member of a set of resources called the primary resource and holds this resource both for a preliminary service time (may possibly be zero) while the customer obtains or uses some sequence of other resources called the secondary subsystem. The two resources may be of completely different types.
This event violates certain assumptions of the product form queueing networks and therefore an easy solution does not exist.

The method of surrogates partitions the queueing delay according to which of the simultaneously held resources is causing it. It then iterates between two models, each of which involves an explicit representation of one of the simultaneously held resource and a "delay server" (with service time but no queueing) acting as a surrogate for queueing delay due to congestion at the other simultaneously held resource.

Kriz [36], has used the decomposition method of Norton's theorem to present a technique for obtaining simple, computationally efficient solution to non-separable queueing networks encountered in the design of symmetric multi-processor systems. The decomposition is assumed to be valid although the local balance is not satisfied. This model also takes into account both primary and secondary resource systems and provides a straight-forward method of finding bounds on performance properties.

Marsan and Gerla [39] have studied the same problem of micro-processor systems design using approximate Markov modeling technique. At first, an exact Markov model
is constructed using a detailed description of state and the method of phases. The size of the state space corresponding to the exact model becomes extremely large as the system size increases, resulting in lengthy calculations. The number of states is therefore reduced by lumping exact states in order to develop approximate models. Lumping is done in such a manner that several states in the exact model are aggregated by one state in the approximate model. This model is assumed to be Markovian although lumping violates some of its assumptions. Based on the hypothesis of a Markovian model, several measures of overall system performance are evaluated. The results, validated by computer simulation experiments, are found to be close bounds on the actual values.

The methods mentioned above have provided a new direction to the research efforts in this area. However, due to lack of generality, these models are not suitable for the CMS problem as seen in Chapter III. A short review of the models currently available for the analysis of CMSs is described below.
2.3 **PREVIOUS WORK RELATED TO COMPUTERISED MANUFACTURING SYSTEMS**

The concept of Computerised Manufacturing has existed now for about fifteen years though its implementation was extremely slow in the first few years of its existence. In the recent years, however, a sudden increase in the number of problems faced by several manufacturing industries in North America has brought the focus of attention of the research and industrial community, to FMS. One of the early descriptions of a FMS can be found in Hutchinson [29].

A number of papers have appeared in literature recently that address the issues of design and operational control of FMSs. Nearly all of this research has been conducted by groups of organised researchers at Texas A & M, Stuttgart, Toronto, Purdue, Draper Labs, MIT and Harvard Universities. A large number of them make use of the closed queueing network model (CAN-Q) by Solberg [54]. The model, which owes its theoretical foundations to Baskett et. al. [11], assumes a finite number of jobs circulating in the network of queues where the customers are the jobs and servers are the machines. This model has a number of restrictive assumptions such as: FIFO queue discipline with exponential service time distribution, same service rate parameter for all classes of customers, no blocking and no machine breakdowns etc., which limit its usefulness. It is
useful, however, in the pre-release planning [see 14] stage and for studying effects of some design variables on system performance.

Another group of researchers at the Purdue University have been studying the operational control aspects of FMSs. A large number of these studies use simulation as the tool for their analysis and rely on real data. Three major problems have been identified by Nof et al. [43]. They are:

1. **The Part-Mix Problem**: Which subset of part types with associated quantities should be produced in each work period?

2. **Process Selection Problem**: If several alternate processes are available to produce a part type, which process should be selected?

3. **The Part Flow Problem**: What should be the sequence of parts flowing into the system and how should parts be allocated to machines within the system?

This study considers some typical situations in a CMS with given production requirements and non-continuous operation using results of simulation experiments. The system performance and machine utilisation are evaluated for a number of various operating rules. The study reveals that the complicated rules relying on dynamic information operate better.
The same problem has been considered by Nof et. al. [44] and a number of different techniques have been used for analysis; for example:

1. GCMS (Generalised Computer Manufacturing System) due to Lenz and Talvage [38].

2. IRHMC (Ingersoll-Rand Heavy Machining Centre Simulator) due to Herald and Nof [26].

3. CAN-Q analytical model due to Solberg [54].

This research sets forth certain guide lines for determining the optimal part-types for concurrent production, part-mix ratio for parts being produced concurrently, and optimal process selection and sequencing policies etc.

Stecke and Solberg [55] and [56] have presented the problem differently and emphasised the existence of two layers in the structure of the problem; namely,

a) Loading problem - allocating operations and tooling to machines, and

b) Real time flow control problem.

This study is also based on simulation experiments. Maximum possible pooling of resources is found to be more efficient. Contrary to the result of Buzacott and Shanthikumar [14], Stecke & Solberg also found that in open networks with different job classes, unbalanced load allocations with more work assigned to the larger group of pooled machines leads to a higher productivity.
Stecke [57] has formulated the production planning problem in FMSs as a Nonlinear Integer Programming problem and developed solution methodologies using several linearization techniques. A sequential analysis of following problems has been suggested for an efficient operation of a FMS and control of issues important to a FMS manager:

1. **Part Type Selection**: What subset of part-types should be immediately and simultaneously processed?

2. **Machine Grouping**: How to partition machines into machining cells so that each machine in a particular group is able to perform the same set of operations?

3. **Production Ratio Problem**: In what ratio, should part types, selected for simultaneous production, be produced?

4. **Resource Allocation**: How to allocate the limited number of pallets and fixtures of each fixture type among the selected part types.

5. **Loading**: How to allocate operations and required tools of the selected part-types among the machine groups subject to capacity and technological limitations of the FMS.

Note that above formulation of the FMS problem is very similar to the one due to Nof et. al. [43]. However, the solution methodologies in the two cases are entirely different. Moreover, Stecke's model is more comprehensive
and includes details like availability of tool changes, set up times etc. etc.

Buzacott and Shanthikumar [14], have proposed several models for determining the production capacity of the Flexible Manufacturing Systems. They have developed a three layered structure of the control system, viz.;

1. **Pre-Release Planning**: at this stage, the objective is to determine the part types to be manufactured, to identify constraints on operations sequence and to determine the operation durations.

2. **Input Control**: at this stage, the sequence and timing of release of jobs to the system are determined.

3. **Operational Control**: at the operational control level, movement of parts between machines is ensured and which of the idle machines should process the next job is decided.

Note that the models by Nof, Barash, Herald & Stecke etc. deal primarily with the input and operational control of item flow in FMSs. The queueing network in [14] is a closed network with a fixed number of jobs circulating in it. All jobs have exponentially distributed service times and the queue discipline at all service stations is FIFO. Each machining cell consists of one machine which can process only one job at a time. The open queueing networks are
approximated by finding the throughput rate - expected no. of departures per unit time - when the number of jobs in the closed system is infinitely large. These models show the desirability of balanced work loads, benefits of diversity in job routing if adequate control on release of jobs exists, and the superiority of common storage over local storage.

Shanthikumar and Sargent [53] have developed a hybrid simulation/analytic model. The value of system throughput is determined once by a simulation experiment and results are used to develop prediction models for system throughput for various levels of loading. Other methods due to Shanthikumar [52] have treated FMS as:

i) A single server queue with service rates equal to the throughput rates of the closed queue.

ii) A network of GI/D/1 queues.

The first approach enables investigation of the effects of various dispatch rules contingent upon the queue of jobs waiting for release. The latter approach can be extended to queue disciplines other than the FIFO, such as the SPT, in order to determine the overall performance of the system. As the number of machines increases, the arrival process at each machine becomes closer to an exponential distribution. Therefore, the latter approach is very suitable for large systems.
In addition to above, there exist a number of models due to Buzacott [13], Kusiak [37], Suri [58] etc. A review of modeling approaches to FMS can be found in Buzacott and Yao [15] and Wilhelm and Sarin [61]. Gupta et. al. [25], have summarised the major modeling approaches in a tabular form and have related functions of various models to their major assumptions, design parameters and solution methodologies. Since in this research we are primarily concerned with the queueing network analysis of computerised manufacturing systems, we shall devote the following few paragraphs to a brief appraisal of the models mentioned above and to a few new techniques to overcome some of their limitations.

A large number of studies in the past have used simulation as a tool for analysing CMSs. While it is possible to construct a model with any desired level of detail using simulation methods, it gives little insight into the processes and design considerations affecting the performance of a FMS. On the other hand, the prevailing analytical models either do not capture the system characteristics well enough or are too complex and computationally intractable to be of any practical significance. On the whole, where simulation has been used for analysis, little efforts have been made to find the theoretical foundations of the results; and where the
treatment is purely theoretical, certain important practical considerations, for example, the materials handling system (MHS), have not been given their due importance in the model.

Yao and Buzacott [63], among others have attempted to relax some of the restrictive assumptions of the earlier models. They consider a central server closed queue network model with multiple servers at each machining cell, finite local buffers resulting in blocking and general service time distributions. The materials handling facility uses specially designed pallets for transportation and the number of jobs in the system is limited by the number of pallets. Solutions to this model are derived through an iterative decomposition approach. This approach combines the classical product form models and some single-node queueing models developed by the same authors [62]. Whenever a product form of the model does not hold, the model is approximated by a product form model using a set of state dependent service rates. In order to find the equivalent state dependent service rates, an iterative scheme is followed until some convergence criterion is satisfied. This model, however, does not give due consideration to contention for materials handling resources as design variables.
In conclusion, the modeling approaches have suffered from being able to address only certain specific and isolated problems related to FMSs design and efficient operation. Since any comprehensive model tends to be computationally complex, active research is needed in the area of developing simplified numerical solution procedures, even at the cost of accuracy, and evolve theoretical formulations using results of computer simulation.
Chapter III
THE CMS MODEL

3.1 INTRODUCTION

The purpose of this chapter is to construct a model of a CMS that realistically captures its important characteristics such as finite local storages and competition among jobs for the AGVs (Automated Guided Vehicles). Most of the previous models that are realistic, tend to be overly complex and intractable. On the other hand, rapid and accurate algorithms are available only for simplistic models that are often far removed from reality. It is important to lay stress on the significance of analytical modeling at this point. There is no doubt, that it is possible to achieve almost any level of detail and a true representation of the CMSs, using techniques of computer simulation, but the procedure is generally very expensive and does not provide sufficient details about the effect of important design variables on the overall system performance. Analytical techniques are comparatively difficult to develop but they can be extremely efficient and accurate. They are well suited for parametric analysis of the computer systems. Moreover, these models reveal important information about
the inner mechanisms of the underlying queuing system that are very useful to designers and engineers.

The first part of this chapter is devoted to a generic description of a CMS with characteristics that cannot be appropriately modeled by the existing techniques. Next, a queuing architecture of this model is formulated. We also examine three sets of currently available solution techniques and their applicability to the model. The three techniques are:

i) Simulation

ii) Exact Analytic Method

iii) Approximate Analytic Method

3.2 CMS ARCHITECTURE

We consider a central server model of a CMS (see fig. 3.1) with machining cells serviced by the Automated Guided Vehicles (AGVs) and robots. The machining cells consist of groups of NC (Numerical Control) machines all under direct computer control. Each machining cell is capable of serving several jobs simultaneously and can perform a number of different machining operations. The machining cells are assumed to have finite capacity local buffers along with the common storage at the central station. This arrangement is found to yield high productivity while maintaining the work-in-process inventory at a low level. The local buffers
smoothen the item flow in the system and reduce machine
down-time. The robots perform all load/unload operations on
the machines and the AGVs and are also responsible for
changes in tools, jigs and fixtures. The importance of AGVs
in a CMS environment rests in their ability to deliver
materials over "flexible routes", interface automatically
with production machines and operate under computer control.

The jobs are serviced in batches that may consist
of different job types. The item inflow to the system is
regulated through a central station. Once a job leaves the
central station, its movements as well as the service
sequence are completely unmanned. Careful planning and
accurate design are therefore crucial for the trouble free
operation of the system. All jobs first undergo a process
called the 'arrival delay' during which the central server
determines their individual service requirements, and inputs
machining data to the DNC machines through the central
computer. The service sequence and routing are determined
by the controlling computer based on dynamic information
such as the status of queues in the network at any time, the
production schedule and the selection criterion etc. The
problem of production planning and routing has been
addressed in [26], [43], [44], [55] & [57] etc. In the
present work we do not address this problem and assume that
an optimal plan exists.
Fig. 3.1 shows a schematic of the flow of jobs through the CMS. After the arrival delay, all jobs queue up at the central station for their turn to be machined, on a FIFO basis. The job at the head of the queue requests for service at one of the work stations capable of performing the service(s) required by its machining sequence. If no such work station is free, then the job waits in a finite capacity local store queue at the first available location.

Before being transported to the machining cell or the local store, the job must acquire one of the free AGVs. A pool of AGVs is available on a FIFO basis for accessing the machining cells. This is the basic instance of simultaneous resource possession. Therefore, we designate machining cells as 'Primary Resources' and the AGVs as 'Secondary Resources'. The jobs, on obtaining a member of the primary resources, must hold this resource while obtaining an AGV in order to be able to access the primary resource for service. The AGVs are required only for transportation and are free to serve other jobs once the transfer has been accomplished. In the event that none of the AGVs is available or all local stores are full, the queue at the central station is blocked. It is assumed that after a service cycle at a work station each job returns to the central store from where it can either request for further service or leave the system to be stored as finished.

* The notations 'local stores' and 'local storages' are used interchangeably throughout this thesis.
Figure 3.1: Schematic diagram of the central server model of a CMS showing item flow.
product at the AS/R (Automated Storage/Retreival) area. The feedback flow is typically handled by a device like a roller conveyor or a carousel.

3.3 A QUEUEING NETWORK MODEL OF THE CMS

The flow of jobs through the CMS model described above could be conveniently represented by a queueing network. For the queueing system, jobs circulating in an associated CMS are 'customers' and machining cells are 'service centres'. The network could then be analysed to determine the blocking, utilisation of servers, average processing delay of the system etc. etc. Note that such attributes of the system can be derived just by knowing some specifics of the item flow and nature of operations. This would be extremely useful to designers, enabling them to study system response to changes in design variables at a very small expense of time and money.

For ease of analysis, we assume that the number of jobs cycling through the CMS model of section 3.2, N, remains constant. In other words, if a job leaves the system to be stored at the AS/R area, another job from the same area takes its place, keeping the total number of jobs in the system constant. An open system can be modeled by allowing N approach infinity. The assumption of a finite customer population is in accordance with the actual CMS operation where jobs are machined in small batches.
The work stations of the CMS are numbered as 1, 2, \ldots, M. The central station is indexed as '0' and has a common storage area, C, large enough to accommodate all the N customers circulating in the system. A schematic of the queueing network is shown in fig. 3.2. The characteristics of this system and assumptions therein are as follows:

i) There are M service stations. The i\textsuperscript{th} service centre is capable of processing \( m_i \) customers at a time. \( m_i \in \{1, 2, \ldots\} \).

ii) Each service station can be treated as an independent unit capable of performing a variety of services.

iii) There are a fixed number of N customers circulating in the system at any time.

iv) There are a total of K units of the secondary resource (AGVs) available to the system.

v) Each customer undergoes an arrival delay (service but no queueing) which may possibly be zero, with a mean equal to \( 1/\lambda \). The arrival delay may have any general distribution described by \( \xi(t) \). For the CMS model, this is the time taken by the central server to program the DNC machines plus the time taken by jobs to travel back to the central station after a service cycle at one of the work stations.

vi) The mean holding time of the secondary resource is equal to \( 1/\mu_t \), with a probability distribution function \( \delta(t) \). For the CMS model, we choose
δ(t) to be a deterministic distribution based upon the experience with AGVs operation.

vii) There are R classes of customers. The processing time of the $i^{th}$ class of customers at the $j^{th}$ service station can have any general distribution function $\gamma_i(t)$, with a mean $= 1/\mu_i,j$. We may assume that the grouping of servers is done such that all service centres can perform nearly all the different operations and serve all classes of customers. Therefore, it will be fair to assume that the mean service time for all customers of class $i$ is $1/\mu_i$ for all service centres. In other words, the service centres may be considered as 'homogeneous servers' leading to a better load balance and higher productivity [14].

viii) A customer can request for service at any one of the service stations. The probability with which the $i^{th}$ customer requests for work station $j$ is $p_{i,j}$; $i = 1, 2, ..., N$ and $j = 1, 2, ..., M$ where:

$$
\sum_{j=1}^{M} p_{i,j} = 1 \quad \forall i.
$$

(3.1)

If the customers can request for any station arbitrarily, and are not biased towards any particular work station, then:

$$
p_{i,j} = 1/M \quad \forall i,j.
$$

(3.2)
ix) At each service station, there is a local store that can hold $Z_i$ customers. This store is only used by customers whose next operation is at station $i$.

x) The time taken by a customer to travel from a service station back to the central station need not be considered explicitly. This is so because, the feedback flow is normally managed by some facility other than the secondary resource subsystem. Typically, this facility may be a roller conveyor or a carousel. Moreover, the arrival delay accounts for the time spent during travel to the central station.

The emerging queueing network, shown in fig. 3.2, represents the CMS quite realistically. It could now be analysed mathematically to study levels of system performance as some of the design variables are varied.
Figure 3.2: Queueing network model of the central station CMS with contention for AGVs: The Secondary Resource.
3.4 AVAILABLE METHODS OF SOLUTION

In this section we discuss the applicability of different techniques available to the systems analyst for analysing the queueing network of fig. 3.2. The available methods could be classified into three major categories: (1) Simulation, (2) Exact Analytic Methods, and (3) Approximate Analytic Methods.

Simulation is easily adaptable to the CMS model. It involves reproduction of exact operation of the CMS on a computer to study its behaviour. Therefore, it requires prohibitively large expense in man-hours and computer time. Besides several practical considerations such as, the number and length of runs required to obtain accurate results, the stopping criterion, etc. still remain unresolved. Moreover simulation can provide only aggregate results of system response and parametric analysis, which is extremely important and necessary for designers in the early stages of the design, is very costly and tedious. We shall, therefore, focus our attention on other methods in order to avoid costly computations.

Exact analytic methods provide accurate results and generally require less computation time than simulation.

* The terms 'analytic' and 'analytical' have been used synonymously throughout this thesis.
With reference to the work of Miller [41] and the general theory of Markov processes, if \( a_{i,j} \) be an element of the transition probability matrix \( A \) of a Markov process, then the stationary state probability vector \( \pi^* \) of the process, if it exists, is a solution to the following system of equations:

\[
\pi = \pi A
\]

(3.3)

\[
\pi e' = 1
\]

(3.4)

\[
\pi_i > 0
\]

(3.5)

where \( e' \) is a column vector of 1's of appropriate size. Hence, the problem is now reduced to that of finding solution to a system of linear equations (3.1) and (3.2) subject to the non-negativity constraint (3.3). However, the Markov property holds only when the service time distributions are exponential and the queue capacities are infinite. Service time distributions other than the exponential can be approximated by using a number of phases of pseudo exponential services [64]. The definition of state space has to be expanded to include information
regarding the number of stages of pseudo service completed. Thus the resulting dimension of state space becomes very large and makes the method cumbersome. Moreover, the number of states grow explosively for any real sized system. With the result, the requirements of computer memory space are large enough to ourstrip any computing system of all available memory; thus making this technique quite impractical.

For the model of the CMS presented here, the number of jobs in the system at any time is limited by the availability of local stores and the number of free AGVs. Consequently, the arrival process is state dependent and the queueing model of section 3.3 does not satisfy local balance. Hence, a relatively easy and inexpensive product form solution does not hold. Therefore, it would be appropriate to conclude that the exact analytic technique available to the systems analyst are not suitable for the CMS model.

Now, we shall discuss some of the approximate analytic methods in the light of the model presented in section 3.3. Several of these procedures have been reviewed in section 2.2. Three of the significant and recent techniques used for analysing multiprocessor systems with shared memories and buses are now examined for applicability.
The method of surrogates described in [32] (see also P. 19) considers two types of service requirements in case of simultaneous possession of resources. First, the non-overlapped service requirement at each primary resource before the secondary subsystem is accessed; and second, an overlapped service requirement during which the primary as well as the secondary resources are held. The simultaneous possession of resources does not occur in this fashion for the CMS model. Also, the method assumes that the service time requirements for each resource type are known and are deterministic. This method lacks generality and does not realistically represent the model of section 3.3. Similarly, for the models described in [36] & [39], (see also P. 20) the global access times that are equivalent to machining times in our model, must be exponential. Moreover, the time spent to establish a link with the global memory is zero and the buses are not set free until the service has been completed on global memories.

In short, none of the known methods is general enough to be suitable for our model. Therefore, we propose a new Semi-Markov modeling method next.
Chapter IV

EXACT AND APPROXIMATE MODELS USING THE SEMI-MARKOV APPROACH

4.1 INTRODUCTION

In this chapter we introduce the well known concept of a Semi-Markov chain to analyse non-separable networks of the type shown in fig. 3.2. We should note that there exist many more real life situations having similar structure of the underlying queueing network. Some of examples of such systems include:

i) Airline reservation system; where the traveler seeking information is a customer and an agent plus a terminal to computer constitute a server. The links provided from each zone to the central information processing unit are the secondary resources in this example.

ii) Direct access storage device (DASD); where the customers are the requests for record(s) and the servers are the individual DASD units. The number of channels to the central unit are the secondary resources.

iii) The multiprocessor systems with shared memories and buses as described in [39], [32] and [36] are also a subset of this class of problems.

- 45 -
Therefore, the applicability and scope of the model presented here are wide ranging.

We present exact as well as approximate analytic models using the Semi-Markov approach. The only restrictive assumption for validity of this model is that the distribution functions of all types of services must be known and must possess rational Laplace transforms. We obtain steady state probabilities of various state occupancies of the stochastic process described by the queueing network of fig. 3.2. Knowing the steady state probabilities, several performance measures of the system can be obtained.

It is observed that the state space grows explosively with increase in the system size for the exact analytic model. This results in very complex and lengthy computations. A considerable reduction in state space is affected by lumping the states of the exact model. Since the transitions between states of the emerging state space do not retain their Markov property, the resulting models are called approximate analytic models. The approximate analysis makes computations considerably simpler and is quite accurate. Some examples of the CMSs are then solved to illustrate the use of this technique.
4.2 THE SEMI-MARKOV APPROACH

A Semi-Markov process is a generalisation of a Markov process. It is a random process whose successive state occupancies are governed by the transition probabilities of a Markov process, but whose stay in any state is described by a random variable that depends on the state presently occupied and on the state to which the next transition will be made [27] & [10]. The process is called discrete-time or continuous-time Semi-Markov process depending on whether the holding time in any state takes on integral values or any positive value, respectively. Since discrete-time Semi-Markov process is a special case of the continuous-time process, we shall restrict our discussion to the latter. Moreover, the continuous-time model is more appropriate for the problems considered in this thesis. Steady state probabilities of any such stochastic process may be found using this method, regardless of the nature of distribution function(s) of the mean wait time(s), while still taking advantage of the simplicity of Markov transitions. This property makes this approach a very powerful tool for analysis. Let;

\[ X_{i,j} = \text{Mean waiting time in state } i \text{ given that the next transition will be made to state } j. \]
\( f_{i,j}(.) = \) Density function of the mean waiting time in state \( i \) given that the next transition will be to state \( j \).

\( a_{i,j} = \) Probability of transition from state \( i \) to state \( j \) or the \( i,j^{th} \) element of the transition probability matrix \( A \).

When the process has just entered a state \( i \), the next state \( j \) is selected according to the transition probability \( a_{i,j} \), but once \( j \) has been selected, the waiting time is specified according to a probability distribution with mean equal to \( X_{i,j} \) and probability density function \( f_{i,j}(.) \). The Markov chain formed by the points of transitions between states is also known as an Imbedded Markov chain. A difficulty arises when we may not be able to define the density function \( f_{i,j}(.) \) in the normal way. This situation appears when a particular value of a random variable can occur with a finite, non-zero probability. The probability density function must be infinite at this point, e.g., in case of a deterministic distribution. We, therefore, expand the concept of density function by first defining a unit impulse function \( \delta(.) \) by:

\[
\delta(t) = \lim_{h \to 0} \begin{cases} 
1/h & \text{if } 0 < t < h \\
0 & \text{if } t > h
\end{cases}, \quad -\infty < t < \infty
\]
Note that the area of the function remains one throughout the limiting process. Then, the symbol used for the contribution from a random variable $x$ that assumes a value $c$ with probability $d$ is, $d \delta(x_0 - c)$.

The joint probability of transition to a state $j$ at time $T$, given that the process entered state $i$ at time zero, $q_{i,j}(T)$ is given by:

$$q_{i,j}(T) = a_{i,j} f_{i,j}(T)$$  \hspace{1cm} (4.2)

The waiting time density function in state $i$ and its mean are given by:

$$W_i(T) = \sum_{j=1}^{n} a_{i,j} f_{i,j}(T) = \sum_{j=1}^{n} q_{i,j}(T)$$  \hspace{1cm} (4.3)

and,

$$\tau_i = \int_{0}^{\infty} \left\{ \sum_{j=1}^{n} a_{i,j} f_{i,j}(t) \right\} \cdot t \, dt$$  \hspace{1cm} (4.4)

where $n = \text{total number of states in the state space.}$
Next, we calculate the interval transition probabilities which are the central statistics for a Semi-Markov process. Let $\phi_{i,j}(T)$ be the probability that a continuous-time Semi-Markov process will be in state $j$ at a time $T$ given that it entered state $i$ at time zero. Now, we can write:

$$\phi_{i,j}(T) = \sum_{k=1}^{n} \int_{0}^{T} q_{i,k}(x) \phi_{k,j}(t-x) \, dx \quad \text{if } i \neq j$$  \hfill (4.5)$$

$$\phi_{i,i}(T) = 1 - \sum_{k=1}^{n} \int_{0}^{T} (1 - \phi_{k,i}(t-x)) q_{i,k}(x) \, dx$$  \hfill (4.6)$$

Above equations are the convolution integrals of the form $\int_{0}^{t} f_{1}(\tau) f_{2}(t-\tau + \cdot) \, d\tau$ which has exponential transforms of type $f_{1}^{e}(s) f_{2}^{e}(s)$. Transformation analysis of (4.5) & (4.6) leads us to the following equation [51]:

$$s \bar{\phi}(s) = [I - \bar{G}(s)]^{-1} [I - \bar{F}_{d}(s)]$$  \hfill (4.7)$$

where $\bar{\phi}(s)$ = the matrix whose $\phi_{i,j}(s)$ term is the Laplace tranform of interval transition probability $\phi_{i,j}(t)$. $\bar{G}(s)$ = the matrix whose $i,j^{th}$ term is $a_{1,j} \bar{F}_{i,j}(s)$.
\( F_d(s) \) = the diagonal matrix whose \( i^{th} \) term is equal to \( \sum_{k=1}^{n} \tilde{F}_{i,k}(s) a_{i,k} \)

\( I \) = the identity matrix of appropriate size.

We can now obtain the limiting (steady state) values of the interval transition probabilities. The procedure for finding these for 'monodesmic' processes is described here. The extension of these ideas to polydesmic processes is straightforward. A stochastic process is called monodesmic when there exists a non-zero transition probability from any state to any other state. Also, a monodesmic Markov process possesses a limiting state probability vector that is independent of the initial state probability vector. All examples cited in this work give rise to monodesmic processes,

\[
\phi = \lim_{T \to \infty} \phi(T) = \lim_{s \to 0} s \phi(s)
\]

(4.8)

From above and on inverting, we obtain:

\[
p_i = \phi_{i,j} = \frac{q_i \tau_i}{\sum_{k=1}^{n} q_k \tau_k} = \frac{q_i \tau_i}{\tau}
\]

(4.9)
where $q_i$ is the limiting state probability of state $i$ for the Imbedded Markov process and the quantity $\bar{t} = \sum_{k=1}^{n} q_k \tau_k$ is the mean time between transitions.

Thus (4.9) gives us a method of finding $p_i$ knowing $q_i$'s and $\tau_i$, which could be derived from equations (3.3), (3.4), (3.5), and (4.4) respectively.

### 4.3 PERFORMANCE MEASURES

Several performance measures of queueing systems may be derived from the knowledge of the steady state probabilities. Let $N_{i,r}$ denote the number of customers of class $r$ being served in a state $i$, then, the expected number of customer being served by the system, $E(N)$, is given by:

$$E(N) = \sum_{i=1}^{n} \left[ \sum_{r=1}^{R} N_{i,r} \right] p_i$$

(4.10)

The average utilisation of the $i^{th}$ server can be calculated using the definition of the state space as follows:

$$\rho_i = 1 - \text{(Prob. that } i^{th} \text{ server is idle)}$$

(4.11)
The throughput of a queueing network is defined as the rate at which the jobs are cycling through the system. Using Little's formula, we can find the throughput of the queueing network by the following equation:

$$TH(N) = \sum_{i=1}^{n} \left( \sum_{r=1}^{R} N_{i,r} \mu_r^* \right) p_i$$

(4.12)

Average delay per arbitrary customer, D, can now be calculated as:

$$D = \left[ N - E(N) \right] / TH(N)$$

(4.13)

Any of the above statistics may be treated as a performance index of an arbitrary system and parametric analysis carried out.

4.4 THE EXACT MODEL

According to this model, the state of the system is described as follows:

$$S = \{ s_1, \theta_1, s_2, \theta_2, \ldots, s_N, \theta_N \}$$

(4.14)

* $\mu_r$ is the rate of service of customers of class r.
where \( s_i \) represents the status of the \( i^{th} \) customer in the system. (customers are indexed as \( 1,2,\ldots,N \)). We assign:

\[
\begin{align*}
    s_i &= \begin{cases} 
        -1 & \text{if the customer is undergoing an arrival delay.} \\
        0 & \text{if the customer is being serviced at the } \theta_i^{th} \\
        j & \text{if the customer is } j^{th} \text{ in the queue waiting for service at the } \theta_i^{th} \\
    \end{cases}
\end{align*}
\]

and,

\[
\theta_i = \text{Number of the station where the } s_i^{th} \text{ customer requests for being served or, is being served, or is awaiting a service.}
\]

This definition describes the system completely and generates a Markov chain at the points of transitions of states. However, the \( 2 \times N \) dimension state definition poses the difficulty of computational complexity. For single job class models, considerable reduction can be affected by lumping states together, resulting in the following definition of the state:

\[
S = \{n_1, n_2, \ldots, n_M, q_0, q_1, q_2, \ldots, q_M, n_k\}
\]

(4.15)
where \( n_i \in [0, 1, 2, \ldots, m_i] \) and represents the number of customers being served at the service station \( i \).

\( q_o \in [0, 1, 2, \ldots, N] \) and represents the number of customers queued at the central service station.

\( q_i \in [0, 1, 2, \ldots, Z_i] \) and represents the number of customers queued at the \( i^{th} \) service station.

\( n_k \in [0, 1, 2, \ldots, K] \) and represents the number of units of the secondary resource free for service.

The representation of states in (4.15) satisfies the sufficiency conditions of lumpability [34]. The following definition of lumpability is taken directly from [34]: "A Markov chain is lumpable with respect to a partition \( A = \{A_1, A_2, \ldots, A_k\} \), if for every starting vector \( \nu \) the lumped process is a Markov chain and the transition probabilities do not depend on the choice of \( \nu \)." When more than one class of customers exist, the imbedded Markov chain is weakly lumpable for a starting vector \( \nu' \), such that there exists an equal probability of finding any job \( i \) at any one of the \( M \) service stations in the system [34]. A transition probability matrix can now be constructed knowing the parameters \( N, M, K, m_i, Z_i, \mu^i, \mu^t, \lambda, \delta(t), \gamma_i(t) \) & \( \xi(t) \) etc., and performance indices evaluated following the results established in sections 4.2 and 4.3.
4.5 THE APPROXIMATE MODELS

The description of state space in (4.15) is very detailed and results in enormously large number of states even for moderately large sized systems. Faster and more efficient models can be formulated by further lumping the states of (4.15) to reduce the size of state space. Two such methods are discussed below.

4.5.1 Model #1

At the first level of approximation, the state space is defined as;

\[ S = \{ n_M, q_0, q_1, q_2, \ldots, q_M, n_k \} \]

(4.16)

where \( n_M \in \{0; 1, 2, \ldots \sum_{i=1}^{M} m_i \} \) and represents the total number of customers being serviced at the M service stations.

\( q_0, q_1, \ldots, q_M \) & \( n_k \) have the same meaning as in (4.15).

The dimension of the states has been reduced from (2M+2) to (M+3) in above formulation. This also results in a phenomenal reduction in the number of states. The lumping of states as in (4.16) satisfies the conditions of weak
lumpability for a starting vector \( \nu' \), described in section 4.4, only for a homogeneous server queueing system with a single job class. An average service rate, \( \mu_{av} \), is used to find transition probabilities, where:

\[
\mu_{av} = \frac{1}{R} [ \mu_1 + \mu_2 + \ldots + \mu_R ]
\]

(4.17)

The system behaviour is analysed under the assumption that the transitions between states still possess the Markov property, though lumpability conditions may not be satisfied. The results of this model are not exact though they are found to be highly accurate for the examples solved here. Again, using equations (3.3), (3.4), (3.5), (4.4), and (4.9) we obtain \( P_i \) and then determine performance indices.

4.5.2 Model #2

Further lumping of states of (4.16) in an attempt to simplify state description even more, leads us to the following definition of state:

\[
S = \{ n_M, q_0, q_M \}
\]

(4.18)
where \( n_M \) and \( q_o \) are the same as in (4.16) and (4.15) respectively.

\[ q_M \in [0,1,2,\ldots, \sum_{i=1}^{M} Z_i] \]

and represents the total number of jobs waiting at the local stores of all \( M \) stations.

Dimension of the state space has now been reduced from \( (M+3) \) to 3. This is a significant reduction. Moreover, the state definition does not expand as the system size increases. Equation (4.18) does not satisfy the conditions of either lumpability or weak lumpability. The results obtained are therefore not exact.

Next we shall solve some examples of CMSs to illustrate the application of the exact and approximate models.

4.6 EXAMPLES & RESULTS

We shall first establish the method of finding the transition probabilities of the imbedded Markov chain for the CMS model discussed in section 3.2. The determination of these probabilities depends on distribution functions \( \delta(t) \), \( \xi(t) \), & \( \gamma_i(t) \). For the purpose of illustration we choose, \( \delta(t) \) to be deterministic, and \( \xi(t) \), \( \gamma_i(t) \), \( \forall \) \( i \), to be exponential distributions. In case of the exact model, for any initial state represented by (4.15), transitions are possible to at most three sets of neighbouring states shown below:
\[ S_1^j = \begin{cases} [n_1, n_2, \ldots, n_{j+1}, n_{j+1}, \ldots, n_m, q_0, q_1, \ldots, q_m, n_k + 1] & \ldots n_j < m_j \\
_1, n_2, \ldots, n_m, q_0, q_1, \ldots, q_{j+1}, q_{j+1}, \ldots, q_m, n_k + 1] & \ldots n_j = m_j \text{ & } q_j \leq z_j \\
_j \in [1, 2, \ldots, M] \text{ & } n_k < k \end{cases} \] (4.19)

\[ s^2 = [n_1, n_2, \ldots, n_m, q_0 - 1, q_1, q_2, \ldots, q_m, n_k - 1] \]

\[ \ldots \vee n_k > 0 \] (4.20)

\[ S_3^j = \begin{cases} [n_1, n_2, \ldots, n_j - 1, n_{j+1}, \ldots, n_m, q_0 + 1, q_1, \ldots, q_m, n_k] & 0 \leq n_j \leq m_j \text{ & } q_j = 0 \\
_1, \ldots, n_m, q_0 + 1, q_1, q_2, \ldots, q_{j-1}, q_{j+1}, \ldots, q_m, n_k] & n_j = m_j \text{ & } q_j > 0 \\
_j \in [1, 2, \ldots, M] \end{cases} \] (4.21)

Transitions to one of the states belonging to either \( S_1^j \), \( S_2 \) or \( S_3^j \) occur when either an AGV completes service, or a job at the central store acquires an AGV or, a job completes machining cycle at one of the machining cells, respectively.

Now,

\[ \delta(t-1/\mu_t) = \begin{cases} 1 & \text{if } 0 < t \leq 1/\mu_t \\
 & \ldots 0 \leq t < \infty \\
0 & \text{if } t > 1/\mu_t \end{cases} \]
(4.22)

\[ \gamma_i(t) = 1 - \exp(-\mu_i t) \]

(4.23)

\[ \xi(t) = 1 - \exp(-\lambda t) \]

(4.24)

Therefore, the transition probabilities are:

(4.25)

\[ a_{s \rightarrow s_1} = \frac{\exp\left(-\left(\mu_1 + \mu_2 + \ldots + \mu_l + q_0 \lambda\right)/n_k \mu_i\right)}{b} \]

(4.26)

\[ a_{s \rightarrow s_2} = q_o \lambda \frac{1 - \exp\left(-\left(\mu_1 + \ldots + \mu_l + q_0 \lambda\right)/n_k \mu_i\right)}{\left(\mu_1 + \ldots + \mu_l + q_0 \right)} \]

(4.27)

\[ a_{s \rightarrow s_3} = \frac{\mu_j \left[1 - \exp\left(-\left(\mu_1 + \ldots + \mu_l + q_0 \lambda\right)/n_k \mu_i\right)\right]}{\left(\mu_1 + \ldots + \mu_l + q_0 \lambda\right)} \]
where \( b \) represents the number of states with at least one machining or local store location free. Mathematically, \( b \in \{1, 2, \ldots, M\} \) such that \( n_i = m_i \) and \( q_i = Z_i \) for \( M-b \) cells,

\[ l \in \{0, 1, 2, \ldots, M\} \] and denotes the number of machining cells such that \( n_l > 0 \). Note that for the present example, the indexing of machining cells is done in the decreasing order of the number of jobs being machined by them,

and \( j \in \{1, 2, \ldots, l\} \).

The unconditional mean waiting time in a state \( i \) may be determined as follows;

\[
\tau_i = \frac{1}{n_k \mu_t} \int_0^{t_i} t \left\{ \frac{\exp(-\mu_1 \cdots + \mu_l + q_0 \lambda)}{n_k \mu_t} \right\} \delta(t - 1/n_k \mu_t) + \frac{\mu_1 \cdots + \mu_l + q_0 \lambda}{n_k \mu_t} \exp(-\mu_1 \cdots + \mu_l + q_0 \lambda) \frac{dt}{n_k \mu_t} \}
\]

[Note that the '+' sign implies that the upper limit of the integration is reached from the positive direction. It follows directly from the definition of a unit impulse function as given by equation (4.1).]

\[ (4.28) \]

We can also establish equations for finding transition probabilities and mean waiting times for the two approximate models in a similar fashion. For the first model, if the state of the system at any time is represented by \((4.16)\), transitions are possible to at most following sets of neighbouring states;
\[ s_1^j = \begin{cases} 
\{ n_M+1, q_0, q_1, \ldots, q_M, n_k+1 \} & \ldots \quad n_M < \sum m_i \\
\{ n_M, q_0, q_1, \ldots, q_j+1, q_j+1, \ldots, q_M, n_k+1 \} & \ldots \quad q_j < z_j \\
\ldots \quad \forall \ j \in [1, 2, \ldots, M] \quad \& \quad n_k < k 
\end{cases} \quad (4.29) \]

\[ s_2 = \{ n_M^*, q_0-1, q_1, \ldots, q_M, n_k-1 \} \quad \ldots \quad n_k > 0 \quad \& \quad q_0 > 0 \quad (4.30) \]

\[ s_3^j = \begin{cases} 
\{ n_M-1, q_0+1, q_1, \ldots, q_M, n_k \} & \ldots \quad n_M > 0 \\
\{ n_M, q_0+1, q_1, \ldots, q_j-1, q_j+1, \ldots, q_M, n_k \} & \ldots \quad q_j > 0 \quad \& \quad n_M > m_j \\
\ldots \quad \forall \ j \in [1, 2, \ldots, M] 
\end{cases} \quad (4.31) \]

Now, the transition probabilities are:

\[ a_{s \rightarrow s_1^j} = 1/b \cdot \left[ \exp \left( - (\mu_1 + \ldots + \mu_i + q_0 \lambda) / n_k \mu_t \right) \right] \quad (4.32) \]

\[ a_{s \rightarrow s_2} = \frac{q_0 \lambda [1 - \exp \left( - (\mu_1 + \ldots + \mu_i + q_0 \lambda) / n_k \mu_t \right)]}{(\mu_1 + \ldots + \mu_i + q_0 \lambda)} \]
\[ a_{s \rightarrow s_3} = \frac{\mu_{av} \left[ 1 - \exp\left( -\left( \mu_1 + \ldots + \mu_l + q_o \lambda \right)/n_k \mu_t \right) \right]}{\left( \mu_1 + \ldots + \mu_l + q_o \lambda \right)} \]

(4.33)

where \( b \) & \( l \) have the same meaning as for the exact model. \( l \) is equal to \( \min\{M, n_d\} \) for this model.

and \( \mu_{av} = (\mu_1 + \ldots + \mu_l)/l \)

Also,

\[ 1/n_k^+ \mu_t \]

\[ T_i = \int_{0}^{t_i} \left\{ \exp\left( -\left( \mu_1 + \ldots + \mu_l + q_o \lambda \right)/n_k \mu_t \right) \right\} \delta\left( t - 1/n_k \mu_t \right) \]

(4.34)

\[ + \left( \mu_1 + \ldots + \mu_l + q_o \lambda \right) \exp\left( -\left( \mu_1 + \ldots + \mu_l + q_o \lambda \right)t/n_k \mu_t \right) \]

(4.35)

Finally for the approximate model #2, if the initial state of the system be represented by (4.18), a transition may occur to at most one of the following sets of neighbouring states;
\[ s_1^j = \begin{cases} \{ n_{M+1}, q_0, q_M \} & \ldots n_M < \sum_{i=1}^{M} z_i \\ \{ n_M, q_0, q_M+1 \} & \ldots n_M > 0 \text{ and } q_M < \sum_{i=1}^{I} z_i \end{cases} \]

\[ \ldots \text{if } n_k < K \] (4.36)

\[ s_2 = \{ n_M, q_0 - 1, q_M \} \quad \ldots n_k > 0 \] (4.37)

\[ s_3^j = \begin{cases} \{ n_{M-1}, q_0+1, q_M \} & \ldots 0 < n_M < \sum_{i=1}^{M} m_i \\ \{ n_M, q_0+1, q_M-1 \} & \ldots 0 < q_M < \sum_{i=1}^{I} z_i \end{cases} \] (4.38)

where \( n_i = K - [N - (n_M + q_0 + q_M)] \) and represents the number of free AGVs as in the earlier models.

\( I \) is the largest integer less than or equal to \( n_M / m_i \)

The transition probabilities may now be calculated as;

\[ a_{s_1^j \rightarrow s_1^j} = \frac{1}{b} \cdot [\exp(- \frac{\mu_{av} + q_0 \lambda}{n_k \mu_t})] \]
\[ a_{s \rightarrow s_2} = \frac{q_o \lambda [1 - \exp(- (\mu_{av} + q_o \lambda) / n_k \mu_t)]}{(\mu_{av} + q_o \lambda)} \]  

\[ a_{s \rightarrow s_3} = \frac{\mu_{av} [1 - \exp(- (\mu_{av} + q_o \lambda) / n_k \mu_t)]}{(\mu_{av} + q_o \lambda)} \]

where \( b \in [1, 2] \) depending on whether only either one of the following conditions is met or both are met simultaneously:

(i) \( n_M < \sum_{i=1}^{M} m_i \)

(ii) \( q_M < \sum_{i=1}^{I} Z_i \)

and \( \mu_{av} = 1/I [\mu_1 + \ldots + \mu_I] \)

The unconditional mean waiting time in an arbitrary state \( i \) may be determined in the same way as in previous models.
Having established a procedure for obtaining \( P_i \)'s and \( r_i \)'s, a benchmark example is then solved with following parameters:

\[
\begin{align*}
N &= 5 \\
M &= 2 \\
m_1 &= m_2 = 1 \\
z_1 &= z_2 = 1 \\
K &= 2 \\
\mu &= 0.5 \text{ jobs/time unit} \\
\lambda &= 1.0 \text{ jobs/time unit} \\
\mu_1 &= \mu_2 = 0.2 \text{ jobs/time unit}
\end{align*}
\]

We also assume,

The state transition diagrams for this example using the exact analytic method, and the approximate models # 1 & 2 shown in figs. 4.1, 4.2, and 4.3. Figure 4.4 shows a schematic of the SLAM [47] network model of this example. Next, we show the how to find transition probabilities and unconditional mean waiting time for this example for an illustrative state using the exact analytic model. From the state indexed '6' in fig. 4.1 \( s_6 = (1,0,3,0,0,1) \), transitions are possible to the following sets of states:

\[
\begin{align*}
S_1 &= \{(1,1,3,0,0,2), (1,0,3,1,0,2)\} \\
S_2 &= \{(1,0,2,0,0,0)\} \\
S_3 &= \{(0,0,4,0,0,1)\}
\end{align*}
\]
and transition probabilities are:

\[ a_{6,8} = a_{6,10} = \frac{1}{2} \left[ \exp\left(-\frac{\mu_1 + 3\lambda}{\mu_t}\right) \right] \]

\[ a_{6,9} = \left(\frac{3\lambda}{\mu_1 + 3\lambda}\right) \left[ 1 - \exp\left(-\frac{\mu_1 + 3\lambda}{\mu_t}\right) \right] \]

\[ a_{6,2} = \left(\frac{\mu_1}{\mu_1 + 3\lambda}\right) \left[ 1 - \exp\left(-\frac{\mu_1 + 3\lambda}{\mu_t}\right) \right] \]

Also,

\[ \tau_6 = \frac{1 - \exp\left(-\frac{\mu_1 + 3\lambda}{\mu_t}\right)}{\left(\mu_1 + 3\lambda\right)} \]

The probabilities of transitions to other states are zero. Above equations can be derived directly from (4.25) through (4.28). The analytic models lend themselves to algorithmic solution procedures where the transition probabilities of the states are also generated by computer using an iterative scheme and equations (4.19) through (4.41).
Figure 4.1: State transition diagram for a 5x2x1 CMS using the exact analytic model.
Figure 4.2: State transition diagram for a $5 \times 2 \times 1$ CMS using the approximate model #1.
Figure 4.3: State transition diagram for a 5x2x1 CMS using the approximate model #2.
Figure 4.4: Schematic of the SLAM network for a 5x2x1 CMS example. (No. of AGVs = 2)
A parametric analysis was then carried out for the benchmark example and the number of jobs in the system was varied from 5 to 30. System processing delay and throughput are treated as performance indices. The results obtained through the three analytic procedures and simulation are tabulated in Table 4.1. It is observed that the two resource subsystems affect the system performance differently for different conditions of load on the system. For the light load conditions, i.e. when N is small, the delay at the central station due to competition among jobs for free AGVs contributes significantly to the system delay. On the other hand, when N is large, or heavy load conditions exist, since $\mu_t > \mu_1 (= \mu_2)$, the blocking is caused only due to local stores being full to capacity. Therefore, the delay caused by contention for secondary resources could be ignored. We have obtained asymptotic bounds on system delay using the three analytic techniques. The lower bounds are obtained by ignoring delay caused by the non-availability of AGVs while the upper bounds are obtained by including the same as a part of the system delay. These bounds along with the actual values obtained through simulation are shown in figs. 4.5, 4.6, and 4.7 for the exact and approximate model #1 & 2 respectively. The actual values lie on upper bounds for light load conditions and on lower bounds for heavier load conditions.
TABLE 4.1

System processing delay and throughput rate as a factor of number of jobs for an "Nx2x1" CMS with 2 AGVs using simulation, the exact model and the approximations 1 & 2.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>No. of Jobs (N)</th>
<th>System Delay (Time Units)</th>
<th>Throughput TH(N)</th>
<th>Jobs/Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Simulation 99% Confidence Interval</td>
<td>Exact Method</td>
<td>Approximate method</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#1</td>
<td>#2</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>16.21 ± 0.331</td>
<td>15.154</td>
<td>16.24</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>18.21 ± 0.37</td>
<td>17.41</td>
<td>19.08</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>21.26 ± 0.412</td>
<td>20.562</td>
<td>22.00</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>23.84 ± 0.455</td>
<td>23.353</td>
<td>24.95</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>26.49 ± 0.545</td>
<td>26.169</td>
<td>27.94</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>29.13 ± 0.545</td>
<td>29.00</td>
<td>30.93</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>42.44 ± 0.78</td>
<td>43.242</td>
<td>42.928</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>54.78 ± 1.02</td>
<td>54.644</td>
<td>54.062</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>67.77 ± 1.25</td>
<td>58.963</td>
<td>69.203</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>81.05 ± 1.46</td>
<td>83.28</td>
<td>83.51</td>
</tr>
</tbody>
</table>
PERFORMANCE MEASURES OF A Nx2x1 CMS WITH 2 AGVs (EXACT ANALYSIS)

+ ASYMPTOTIC BOUNDS BY EXACT ANALYSIS
O ACTUAL VALUE BY SIMULATION

FIG. 4.5 ASYMPTOTIC BOUNDS ON SYSTEM DELAY USING EXACT MODEL
PERFORMANCE MEASURES OF A Nx2x1 CMS WITH 2 AGVS (APPROX MODEL #1)

ASYMPTOTIC BOUNDS BY APPROX MODEL #1
○ ACTUAL VALUE BY SIMULATION

FIG. 4.6 ASYMPTOTIC BOUNDS ON SYSTEM DELAY USING APPROX MODEL #1
PERFORMANCE MEASURES OF A Nx2x1 CMS
WITH 2 AGVs (APPROX MODEL #2)

+ ASYMPTOTIC BOUNDS BY APPROX MODEL #2
• ACTUAL VALUE BY SIMULATION

FIG. 4.7 ASYMPTOTIC BOUNDS ON SYSTEM DELAY USING APPROX MODEL #2
The results of computer simulation are found to exhibit small fluctuations in the values of performance indices even after lengthy runs. Therefore, a 99% confidence interval on the mean system delay is calculated under the assumption of normality which follows directly from the law of large numbers. The results obtained through the three analytical techniques and the 99% confidence limits are shown in figs. 4.8, 4.9, and 4.10. The analytical results are found to lie within the confidence interval. System throughput is plotted against the number of jobs in the system using all four techniques discussed here as shown in fig. 4.11. The results of approximations #1 & 2 are found to be lower and upper bounds respectively on the system throughput. The upper bound in case of model #2 can be explained intuitively. Lumping of local store queues and service capacities at the machining cells leads to a random redistribution of jobs among the machining locations resulting in a better utilisation of machines.

A similar parametric analysis may be carried out with respect to any one of the parameters that define the system. Typical CPU time requirements for the benchmark example on an IBM 3031 computer are (i) 75 seconds, when using the simulation; (ii) 8.5 seconds, when using the exact model, (iii) 7.5 seconds, when using the approximate model #1, (iv) and 5.5 seconds, when using the approximate model
PERFORMANCE MEASURES OF A Nx2x1 CMS
WITH 2 AGVs - THE EXACT MODEL

+ SIMULATION RESULTS - 99% C.I.
○ THE EXACT MODEL RESULTS

FIG. 4.8 99% CONFIDENCE INTERVAL ON RESULTS OF SIMULATION &
RESULTS OF THE EXACT MODEL
PERFORMANCE MEASURES OF A Nx2x1 CMS
WITH 2 AGVs - APPROXIMATE MODEL #1

+ SIMULATION RESULTS - 99% C.I.
O APPROX. MODEL #1 RESULTS

FIG. 4.9 99% CONFIDENCE INTERVAL ON RESULTS OF SIMULATION &
RESULTS OF APPROXIMATE MODEL #1
PERFORMANCE MEASURES OF A \( N \times 2 \times 1 \) CMS
WITH 2 AGVs - APPROXIMATE MODEL #2

+ SIMULATION RESULTS - 99% C.I.

○ APPROX. MODEL #2 RESULTS

FIG. 4.10 99% CONFIDENCE INTERVAL ON RESULTS OF SIMULATION &
RESULTS OF APPROXIMATE MODEL #2
PERFORMANCE EVALUATION OF A Nx2x1 CMS WITH 2 AGVs

FIG. 4.11 SYSTEM THROUGHPUT AS A FACTOR OF NO. OF JOBS
Chapter V

SUMMARY, CONCLUSIONS & FUTURE SCOPE

This thesis has presented quantitative performance models of queueing networks encountered in Computerised Manufacturing Systems (CMSs) using Semi-Markov approach. Computational results of several examples, using the models developed, have also been presented.

It has been observed in Chapter II that many solution methods are valid only under the assumptions of either (i) Markovian networks or (ii) product form and local balance [19]. Some approximate solutions dedicated to solving multiprocessor systems with shared resources ([32], [39] & [36]) have also been discussed. Models of CMSs have been found to be lacking in detail or computational simplicity. Although, the materials handling equipment forms the core of CMSs by linking various machines and sequencing operations in an unmanned environment, it had not received much attention in the past.

Chapter III has presented a model that captures the important features of a CMS, e.g., limited local buffers, limited materials handling resources and batch
processing etc. The realistic CMS performance model is based upon case studies and descriptions of existing CMSs, e.g., [1], [2], [9], & [29] etc. The model has then been applied to a network of queues taking into account the difference between the primary resources (machining centres) and the secondary resources (AGVs).

A Semi-Markov modeling approach has been proposed to solve the general queueing system in chapter IV. The queueing system could be used to model many real life computer controlled systems with proper choice of parameters. The Semi-Markov modeling technique has been reduced to solving a set of simple linear equations. The procedure developed could provide a tool to the analysts to obtain an exact solution with any type of distribution at any service centre. The only restriction is that the service-time distribution of all services must have rational Laplace transforms. Next, two approximate models have been proposed, also using the Semi-Markov approach, that simplify computations. A benchmark example of Nx2x1 CMS with 2 AGVs has been analysed in detail. All computational results have been validated by comparing them with simulation results using SLAM processor [47]. Parametric analysis has also been carried out to investigate the sensitivity of the results. The approximate results have been found to be quite close to exact results either when the system size is
large or when the system load is moderately heavy. The results of approximate models 1 & 2 have been observed to be the lower and upper bounds respectively on the system throughput value obtained by using the exact model. The bounds have been found to be quite tight (typical % errors being well within 10%) on system performance indices.

The Semi-Markov modeling approach is directly applicable to any queueing system with general service-time distributions and competition among customers for primary as well as secondary type resources. It is efficient as it exploits the Markovian nature of transitions between states. It offers accurate analytical models of systems with general distributions of service durations and provides computational simplicity.

The models presented here could be used for various computer systems, some of which have been described in section 4.1, having an underlying queueing network similar to the one described in this thesis. An important area of further research is the generalisation of this model to include more than one type of secondary resources. The model developed here could also be extended to include priorities in service, and jockeying and balking of customers.
Appendix A

EXAMPLE OF THE SLAM NETWORK COMPUTER PROGRAM

I  COMPUTER LISTING OF SLAM NETWORK
   PROGRAM FOR 5x2x1 CMS WITH 2 AGVs.

II SAMPLE OUTPUT OF SLAM SUMMARY
   REPORT FOR THE 5x2x1 CMS EXAMPLE.
A.1 COMPUTER LISTING OF SLAM NETWORK PROGRAM

//JOENAME JOB (XXXX,XXX,2,3),'D. GUPTA',CLASS=A,MSGLEVEL=(1,1)
// EXEC SLAMG
//GO SYSIN DD *
GEN,D.GUPTA,MASTER THESIS,03/10/84,1;
LIMIT,6,10,100;
NETWORK;

FOR THIS EXPERIMENT, THE CMS HAS 5 JOBS, 2 AGVS AND LOCAL
STORAGE SPACE FOR 1 JOB AT EACH OF 2 M/CING CELLS.
RES1 = TOTAL NUMBER OF UNITS OF LOCAL STORAGE SPACE AVAILABLE
AT M/CING CELL #1
RES2 = TOTAL NUMBER OF UNITS OF LOCAL STORAGE SPACE AVAILABLE
AT M/CING CELL #2
AGV = TOTAL # OF AUTOMATED GUIDED VEHICLES AVAILABLE
STN0 REFERS TO THE CENTRAL LOAD/UNLOAD STATION
STN1 REFERS TO THE M/CING CELL #1
STN2 REFERS TO THE M/CING CELL #2
CREATE RESOURCE BLOCKS FOR LOCAL BUFFER AND NUMBER OF
AUTOMATED GUIDED VEHICLES AVAILABLE
RESOURCE/LATHS(2),1;
RESOURCE/MILLS(2),2;
RESOURCE/AGV(2),3;
CREATE,1,0,5; INITIAL ENTRIES AT THE CENTRAL STORAGE
ACT...,SINO;

SINO ASSIGN,ATRIB(1)=TNOW,1;
ACT,EXPON(1..3),0.5,RES1; REQUEST FOR RES#1 WITH PROB 0.5
AND ARRIVAL DELAY OF 1.0
ACT,EXPON(1..3),0.5,RES2; REQUEST FOR RES#2 WITH PROB 0.5
AND ARRIVAL DELAY OF 1.0
RES1 AWAIT(1),LATHS/1; WAIT IN QUEUE 1 UNTIL RES#1 IS AVAILABLE
ASSIGN,ATRIB(2)=1.;
ACT...,RES3;
RES2 AWAIT(2),MILLS/1; WAIT IN QUEUE 1 UNTIL RES#2 IS AVAILABLE
ASSIGN,ATRIB(2)=2.;
ACT...,RES3;
RES3 AWAIT(3),AGV/1; WAIT IN QUEUE 1 IF AGV IS NOT AVAILABLE
ACT(1)/1,2.; TRAVEL TIME FOR AGV IS 2 MINUTES
FREE,AGV/1; SET THE AGV FREE AFTER SERVICE
ACT, , ATRIB(2), EQ.1, , STN1;
ACT, , ATRIB(2), EQ.2, , STN2;
STN1 QUEUE(4); QUEUE FOR M/C CELL OPERATION, MAX CAP.=1
ACT(1)/2, EXPON(5, 1);
FREE, LATHS/1, 1; SET ONE LOCAL BUFFER SPACE FREE AFTER SERVICE
ACT, , BEE;
STN2 QUEUE(5); QUEUE FOR M/C CELL OPERATION, MAX CAP.=1
ACT(1)/3, EXPON(5, 2);
FREE, MILLS/1, 1; SET ONE LOCAL BUFFER SPACE FREE AFTER SERVICE
BEE COLCT, INT(1), SYSTEM TIME, 20/10/5, 1; COLLECT STATISTICS ON
; THRUPUT TIME FOR ANY JOB
ACT, , STNO;
ENDNETWORK;
INIT, 0, 20000;
MONTR, SUMRY, 200, 10000;
FIN;
/*
//
A.2 SAMPLE OUTPUT OF SLAM SUMMARY REPORT

SLAM SUMMARY REPORT

SIMULATION PROJECT MASTER THESIS
BY D. GUPTA

DATE 3/10/1984
RUN NUMBER 1 OF 1

CURRENT TIME 0.2000E+05
STATISTICAL ARRAYS CLEARED AT TIME 0.0

**STATISTICS FOR VARIABLES BASED ON OBSERVATION**

<table>
<thead>
<tr>
<th>MEAN VALUE</th>
<th>STANDARD DEVIATION</th>
<th>COEFF. OF VARIATION</th>
<th>MINIMUM VALUE</th>
<th>MAXIMUM VALUE</th>
<th>NUMBER OF OBSERVATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM TIME</td>
<td>0.1621E 02</td>
<td>0.1037E 02</td>
<td>0.6398E 00</td>
<td>0.2043E 01</td>
<td>0.7479E 02</td>
</tr>
</tbody>
</table>

**FILE STATISTICS**

<table>
<thead>
<tr>
<th>FILE NUMBER</th>
<th>ASSOCIATED NODE TYPE</th>
<th>AVERAGE LENGTH</th>
<th>STANDARD DEVIATION</th>
<th>MAXIMUM LENGTH</th>
<th>CURRENT LENGTH</th>
<th>AVERAGE WAITING TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AWAIT</td>
<td>0.8534</td>
<td>1.0355</td>
<td>3</td>
<td>0</td>
<td>5.5306</td>
</tr>
<tr>
<td>2</td>
<td>AWAIT</td>
<td>0.7778</td>
<td>1.0054</td>
<td>5</td>
<td>3</td>
<td>5.0489</td>
</tr>
<tr>
<td>3</td>
<td>AWAIT</td>
<td>0.0400</td>
<td>0.2133</td>
<td>2</td>
<td>0</td>
<td>0.1298</td>
</tr>
<tr>
<td>4</td>
<td>QUEUE</td>
<td>0.4554</td>
<td>0.4980</td>
<td>1</td>
<td>0</td>
<td>2.9512</td>
</tr>
<tr>
<td>5</td>
<td>QUEUE</td>
<td>0.4244</td>
<td>0.4942</td>
<td>1</td>
<td>1</td>
<td>2.7575</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>3.4474</td>
<td>1.0081</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1.0099</td>
</tr>
</tbody>
</table>
**REGULAR ACTIVITY STATISTICS**

<table>
<thead>
<tr>
<th>ACTIVITY INDEX</th>
<th>AVERAGE UTILIZATION</th>
<th>STANDARD DEVIATION</th>
<th>MAXIMUM UTILIZATION</th>
<th>CURRENT UTILIZATION</th>
<th>ENTITY COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6164</td>
<td>0.7460</td>
<td>2</td>
<td>0</td>
<td>6164</td>
</tr>
</tbody>
</table>

**SERVICE ACTIVITY STATISTICS**

<table>
<thead>
<tr>
<th>ACTIVITY INDEX</th>
<th>START NODE LABEL/TYP</th>
<th>SERVER CAPACITY</th>
<th>AVERAGE UTILIZATION</th>
<th>STANDARD DEVIATION</th>
<th>CURRENT UTILIZATION</th>
<th>AVERAGE BLOCKAGE</th>
<th>MAXIMUM IDLE TIME/SERVERS</th>
<th>MAXIMUM BUSY TIME/SERVERS</th>
<th>ENTITY COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>STN1 QUEUE</td>
<td>1</td>
<td>0.7747</td>
<td>0.4178</td>
<td>0</td>
<td>0.0</td>
<td>65.3750</td>
<td>124.6094</td>
<td>3086</td>
</tr>
<tr>
<td>3</td>
<td>STN2 QUEUE</td>
<td>1</td>
<td>0.7532</td>
<td>0.4311</td>
<td>1</td>
<td>0.0</td>
<td>65.0313</td>
<td>125.8750</td>
<td>3076</td>
</tr>
</tbody>
</table>

**RESOURCE STATISTICS**

<table>
<thead>
<tr>
<th>RESOURCE NUMBER</th>
<th>RESOURCE LABEL</th>
<th>CURRENT CAPACITY</th>
<th>AVERAGE UTILIZATION</th>
<th>STANDARD DEVIATION</th>
<th>MAXIMUM UTILIZATION</th>
<th>CURRENT UTILIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LATHS</td>
<td>2</td>
<td>1.5599</td>
<td>0.7097</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>MILLS</td>
<td>2</td>
<td>1.5041</td>
<td>0.7353</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>AGV</td>
<td>2</td>
<td>0.6164</td>
<td>0.7460</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESOURCE NUMBER</th>
<th>RESOURCE LABEL</th>
<th>CURRENT AVAILABLE</th>
<th>AVERAGE AVAILABLE</th>
<th>MINIMUM AVAILABLE</th>
<th>MAXIMUM AVAILABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LATHS</td>
<td>2</td>
<td>0.4400</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>MILLS</td>
<td>0</td>
<td>0.4959</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>AGV</td>
<td>2</td>
<td>1.3836</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

-90-
**HISTOGRAM NUMBER: 1**

**SYSTEM TIME**

<table>
<thead>
<tr>
<th>OBSV FREQ</th>
<th>RELA FREQ</th>
<th>CUML FREQ</th>
<th>UPPER CELL LIMIT</th>
<th>0</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>0.324</td>
<td>0.324</td>
<td>0.1000E 02</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>1353</td>
<td>0.220</td>
<td>0.544</td>
<td>0.1500E 02</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>1036</td>
<td>0.168</td>
<td>0.712</td>
<td>0.2000E 02</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>655</td>
<td>0.106</td>
<td>0.818</td>
<td>0.2500E 02</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>454</td>
<td>0.074</td>
<td>0.892</td>
<td>0.3000E 02</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>293</td>
<td>0.048</td>
<td>0.940</td>
<td>0.3500E 02</td>
<td>**</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>181</td>
<td>0.029</td>
<td>0.969</td>
<td>0.4000E 02</td>
<td>**</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>90</td>
<td>0.015</td>
<td>0.984</td>
<td>0.4500E 02</td>
<td>**</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>49</td>
<td>0.008</td>
<td>0.992</td>
<td>0.5000E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>20</td>
<td>0.003</td>
<td>0.995</td>
<td>0.5500E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>17</td>
<td>0.003</td>
<td>0.998</td>
<td>0.6000E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>0.002</td>
<td>0.999</td>
<td>0.6500E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>0.000</td>
<td>1.000</td>
<td>0.7000E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>0.000</td>
<td>1.000</td>
<td>0.7500E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.8000E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.8500E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.9000E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.9500E 02</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.1000E 03</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.1050E 03</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>0.1100E 03</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1.000</td>
<td>INF</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**STATISTICS FOR VARIABLES BASED ON OBSERVATION**

<table>
<thead>
<tr>
<th>MEAN VALUE</th>
<th>STANDARD DEVIATION</th>
<th>COEFF. OF VARIATION</th>
<th>MINIMUM VALUE</th>
<th>MAXIMUM VALUE</th>
<th>NUMBER OF OBSERVATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM TIME</td>
<td>0.1621E 02</td>
<td>0.1037E 02</td>
<td>0.6398E 00</td>
<td>0.2043E 01</td>
<td>0.7479E 02</td>
</tr>
</tbody>
</table>

-91-
Appendix B

COMPUTER PROGRAMS FOR THE ANALYTICAL MODELS

I  SAMPLE PROGRAM FOR THE EXACT MODEL
II  SAMPLE PROGRAM FOR THE APPROX MODEL#1
III  SAMPLE PROGRAM FOR THE APPROX MODEL#2
B.1 SAMPLE PROGRAM FOR THE EXACT MODEL

//JOBNANE JOB (XXXX, XXX), 'D. GUPTA', CLASS=A
//FVRTVCLO EXEC FORTGDLG
//FORT.SYSIN DD *
C ****************************************************************************************************************************
C THIS PROGRAMME CALCULATES THE STEADY STATE PROBABILITIES *
C AND SEVERAL PERFORMANCE MEASURES KNOWING THE TRANSITION RATE *
C MATRIX. *
C ****************************************************************************************************************************
DIMENSION A(50,50), B(50), X(25), WA(25), PI(25), WK(1300), C(4)
1, Q(3)
INTEGER INK(25)
REAL LD, MT, MI, MII
C ****************************************************************************************************************************
C THE INPUT PARAMETERS ARE DEFINED AS FOLLOWS: *
C A = THE M BY N COEFFICIENT MATRIX. *
C B = THE M BY NB MATRIX OF RIGHT HAND SIDES. *
C C = COMMUNICATION VECTOR OF LENGTH 4 (INPUT IF IND=1, AND OUTPUT)*
C M = THE NUMBER OF ROWS IN A AND B. *
C N = THE NUMBER OF COLUMNS IN A AND ROWS IN X. (INPUT) *
C NB = NUMBER OF RIGHT HAND SIDES. (COLUMNS IN B). (INPUT) *
C IA = ROW DIMENSION OF MATRIX A AS SPECIFIED IN THE CALLING PROG *
C IB = ROW DIMENSION OF MATRIX B AS SPECIFIED IN THE CALLING PROG *
C IX = ROW DIMENSION OF MATRIX X AS SPECIFIED IN THE CALLING PROG *
C IND = OPTION SELECTION INDICATOR (DEFAULT=0) *
C NS = 5 = THE NUMBER OF JOBS IN THE SYSTEM. *
C K = THE NUMBER OF AGVS. *
C IZ = LOCAL STORAGE SPACES AVAILABLE AT EACH MACHINING CELL. *
C LD = ARRIVAL DELAY AT THE CENTRAL STATION *
C MT = 0.5 = THE RATE AT WHICH JOBS ARE TRANSFERRED BY THE AGVS *
C MI = 0.2 = THE MACHINING RATE AT CELL # 1 *
C MII = 0.2 = THE MACHINING RATE AT CELL, #2 *
C ****************************************************************************************************************************
IA=50
IB=50
READ(5,1) M, N, NB, IND, IX, NS, K, IZ, LD, MT, MI, MII
1 FORMAT(8I3, 5X, 4F6.2)
READ(5,2) C(1),C(2),C(3)
2 FORMAT(3F10.2)
   DO 5 I=1,M
   DO 5 J=1,N
   A(I,J)=0.0
   IF(I.EQ.M) A(I,J)=1.0
   IF(I.EQ.J) A(I,J)=-1.0
5 CONTINUE
   A(2,1)=1.0
   A(5,2)=1-EXP(-(NS-1)*LD/MT)
   A(3,2)=(1/2)*EXP(-(NS-1)*LD/MT)
   A(4,2)=(1/2)*EXP(-(NS-1)*LD/MT)
   A(1,3)=M/((Mi+(NS-1)*LD)
   A(6,3)=((NS-1)*LD)/(Mi+(NS-1)*LD)
   A(7,4)=((NS-1)*LD)/(Mi+(NS-1)*LD)
   A(1,4)=Mi/((Mi+(NS-1)*LD)
   A(6,5)=0.50
   A(7,5)=0.50
   A(2,6)=Mi*(1-EXP(-(Mi+(NS-2)*LD)/MT))/((NS-2)*LD+Mi)
   A(9,6)=((NS-2)*LD*(1-EXP(-(Mi+(NS-2)*LD)/MT))/((NS-2)*LD+Mi)
   A(8,6)=(1/2)*EXP(-(Mi+(NS-2)*LD)/MT).
   A(10,6)=(1/2)*EXP(-(Mi+(NS-2)*LD)/MT)
   A(2,7)=Mi*(1-EXP(-(Mi+(NS-2)*LD)/MT))/((NS-2)*LD+Mi)
   A(11,7)=((NS-2)*LD*(1-EXP(-(Mi+(NS-2)*LD)/MT))/((NS-2)*LD+Mi)
   A(8,7)=(1/2)*EXP(-(Mi+(NS-2)*LD)/MT)
   A(12,7)=(1/2)*EXP(-(Mi+(NS-2)*LD)/MT)
   A(3,8)=Mi/((Mi+Mi+(NS-2)*LD)
   A(4,8)=Mi/((Mi+Mi+(NS-2)*LD)
   A(13,8)=((NS-2)*LD)/((Mi+Mi+(NS-2)*LD)
   A(5,9)=1-EXP(-Mi/((2*MT))
   A(13,9)=(1/2)*EXP(-Mi/((2*MT))
   A(14,9)=(1/2)*EXP(-Mi/((2*MT))
   A(3,10)=Mi/((Mi+(NS-2)*LD)
   A(14,10)=((NS-2)*LD)/((Mi+(NS-2)*LD)
   A(5,11)=1-EXP(-(Mi/((2*MT))
   A(13,11)=(1/2)*EXP(-(Mi/((2*MT))
   A(15,11)=(1/2)*EXP(-(Mi/((2*MT))
   A(4,12)=Mi/((Mi+(NS-2)*LD)
   A(15,12)=((NS-2)*LD)/((Mi+(NS-2)*LD)
   A(6,13)=Mi*(1-EXP(-(Mi+Mi+(NS-3)*LD)/MT))/((Mi+Mi+(NS-3)*LD)
   A(7,13)=Mi*(1-EXP(-(Mi+Mi+(NS-3)*LD)/MT))/((Mi+Mi+(NS-3)*LD)
   A(16,13)=((NS-3)*LD*(1-EXP(-(Mi+Mi+(NS-3)*LD)/MT))/((Mi+Mi+
   A(17,13)=(1/2)*EXP(-(Mi+Mi+(NS-3)*LD)/MT)
   A(19,13)=(1/2)*EXP(-(Mi+Mi+(NS-3)*LD)/MT)
   A(6,14)=Mi*(1-EXP(-(Mi+(NS-3)*LD)/MT))/((Mi+(NS-3)*LD)
   A(18,14)=((NS-3)*LD*(1-EXP(-(Mi+(NS-3)*LD)/MT))/((Mi+(NS-3)*LD)
   A(17,14)=EXP(-(Mi+(NS-3)*LD)/MT)
A(7,15)=MII*(1-EXP(-(MII+(NS-3)*LD)/MT))/(MII+(NS-3)*LD)
A(20,15)=(NS-3)*LD*(1-EXP(-(MII+(NS-3)*LD)/MT))/(MII+(NS-3)*LD)
A(19,15)=EXP(-(MII+(NS-3)*LD)/MT)
A(11,16)=MII*(1-EXP(-(MII+MII)/(2*MT)))/(MII+MII)
A(9,16)=MII*(1-EXP(-(MII+MII)/(2*MT)))/(MII+MII)
A(21,16)=(1/2)*EXP(-(MII+MII)/(2*MT))
A(22,16)=(1/2)*EXP(-(MII+MII)/(2*MT))
A(10,17)=MII/(MII+MII+(NS-3)*LD)
A(8,17)=MII/(MII+MII+(NS-3)*LD)
A(21,17)=(NS-3)*LD/(MII+MII+(NS-3)*LD)
A(9,18)=1-EXP(-MII/(2*MT))
A(21,18)=EXP(-MII/(2*MT))
A(8,19)=MII/(MII+MII+(NS-3)*LD)
A(12,19)=MII/(MII+MII+(NS-3)*LD)
A(22,19)=(NS-3)*LD/(MII+MII+(NS-3)*LD)
A(11,20)=1-EXP(-MII/(2*MT))
A(22,20)=EXP(-MII/(2*MT))
A(14,21)=MII*(1-EXP(-(MII+MII)/MT))/(MII+MII)
A(13,21)=MII*(1-EXP(-(MII+MII)/MT))/(MII+MII)
A(23,21)=EXP(-(MII+MII)/MT)
A(13,22)=MII*(1-EXP(-(MII+MII)/MT))/(MII+MII)
A(15,22)=MII*(1-EXP(-(MII+MII)/MT))/(MII+MII)
A(23,22)=EXP(-(MII+MII)/MT)
A(17,23)=MII/(MII+MII)
A(19,23)=MII/(MII+MII)

PRINT10
10 FORMAT('1', 'THE FOLLOWING MATRIX REPRESENTS THE TRANSITION RATE MATRIX: ')
DO 15 J=1,N
DO 15 I=1,M
PRINT20, I, J, A(I, J)
20 FORMAT(5X, 'A(', I2, ',', I2, ', ') = ', F12.6)
15 CONTINUE
DO 25 I=1, M
B(I) = 0.0
25 CONTINUE
B(M) = 1.0
C**** CALCULATE THE STEADY STATE PROBABILITIES OF IMBEDDED MARKOV CHAIN CALL LLQBF (A, IA, M, N, B, IB, NB, IND, C, X, IX, IWK, WK, IER)
DO 30 I=1, N
PRINT31, I, X(I)
31 FORMAT(5X, 'THE OUTPUT PROBABILITIES OF IMBEDDED MARKOV CHAIN ARE ')
l = X(',I2,') = ' ,F10.6)

30 CONTINUE

C**** MEAN RESIDENCY TIMES IN EACH STATE ARE CALCULATED AS FOLLOWS:

WA(1) = 1/(NS*LD)
WA(2) = (1-EXP(-(NS-1)*LD/MT))/((NS-1)*LD)
WA(3) = 1/(MII+(NS-1)*LD)
WA(4) = 1/(MI+MI+(NS-1)*LD)
WA(5) = 1/(2*MT)
WA(6) = (1-EXP(1-((NS-2)*LD+MI)/MT))/((MI+NS-2)*LD)
WA(7) = (1-EXP(1-((NS-2)*LD+MI)/MT))/((MI+NS-2)*LD)
WA(8) = 1/(MII+MI+(NS-2)*LD)
WA(9) = (1-EXP(1-((MI+2)*MT))/MI
WA(10) = 1/(MI+MI+(NS-2)*LD)
WA(11) = (1-EXP(1-((MI+2)*MT))/MI
WA(12) = 1/(MI+MI+(NS-2)*LD)
WA(13) = (1-EXP(1-((NS-3)*LD+MI+MI)/MT))/((MI+MI+NS-3)*LD)
WA(14) = (1-EXP(1-((NS-3)*LD+MI+MI)/MT))/((MI+NI+NS-3)*LD)
WA(15) = (1-EXP(1-((NS-3)*LD+MI+MI)/MT))/((MI+MI+NS-3)*LD)
WA(16) = (1-EXP(1-((MI+MI+NS-3)*LD)
WA(17) = 1/(MI+MI+MI+(NS-3)*LD)
WA(18) = (1-EXP(1-((MI+2)*MT))/MI
WA(19) = 1/(MI+MI+MI+(NS-3)*LD)
WA(20) = (1-EXP(1-((MI+2)*MT))/MI
WA(21) = (1-EXP(1-((MI+MI+MI)/MT))/((MI+MI+MI)
WA(22) = (1-EXP(1-((MI+MI+MI)/MT))/((MI+MI+MI)
WA(23) = 1/(MI+MI+MI)

C**** CALCULATE THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV
C QUENING PROCESS.

SUM = 0.0
DO 40 I=1,N
SUM = SUM + WA(I)*X(I)
40 CONTINUE

DO 50 II=1,N
FII(II) = WA(II)*X(II)/SUM
50 CONTINUE

PRINT 55
55 FORMAT('1',2X,'THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV P
1RCESS ARE:')
DO 65 I=1,N
PRINT 60, I, I, FI(I)
60 FORMAT('0',5X,'PROB. THAT THE SYSTEM STATE IS ''I2,'',
1''FI('',I2,'') = ',' ,F12.6)
65 CONTINUE
C *******************************************************
C THE FOLLOWING SEGMENT OF THE PROGRAMME IS TO CALCULATE THE
C PERFORMANCE MEASURES;
C Q(I) = EXPECTED QUEUE LENGTH AT STATION I.
C \( Q = \) EXPECTED SYSTEM DELAY FOR AN ARBITRARY CUSTOMER
C ZETA = THRUPUT RATE OF THE COMPUTERISED MANUF SYSTEM
C R1 = AVERAGE UTILISATION OF M/CING CELL#1
C R2 = AVERAGE UTILISATION OF M/CING CELL#2
C *******************************************************
Q(1)=NS*(PI(1)+PI(2)+PI(5)) + (NS-1)*(PI(3)+PI(4)+PI(6)+PI(7)
1+PI(9)+PI(11)) + (NS-2)*(PI(8)+PI(10)+PI(12)+PI(13)+PI(14)+PI(15)
1+PI(16)+PI(18)+PI(20)) + (NS-3)*(PI(17)+PI(19)+PI(21)+PI(22)+PI(123))
Q(2)= 2*(PI(10)+PI(14)+PI(17)+PI(18)+PI(21)+PI(23)) +(PI(3)+PI(6)
1+PI(8)+PI(9)+PI(13)+PI(16)+PI(19)+PI(22))
Q(3)= 2*(PI(12)+PI(15)+PI(20)+PI(19)+PI(22)+PI(23)) +(PI(4)+PI(7)
1+PI(8)+PI(11)+PI(13)+PI(16)+PI(17)+PI(21))
RO= 0.5*(PI(2)+PI(6)+PI(7)+PI(13)+PI(14)+PI(15)+PI(21)+PI(22))
1+PI(5)+PI(9)+PI(11)+PI(16)+PI(18)+PI(20)
ZET1=RO*MT
R1= 1-(PI(1)+PI(2)+PI(4)+PI(5)+PI(7)+PI(11)+PI(12)+PI(15)+PI(20))
R2= 1-(PI(1)+PI(2)+PI(3)+PI(5)+PI(6)+PI(9)+PI(10)+PI(14)+PI(18))
ZETA =R1+R2
\( Q = \) Q(1)/ZETA + (Q(2)+Q(3))/(ZET1)
DO 66 I=1,3
IFL=I-1
PRINT70, IFL, IFL, Q(I)
66 CONTINUE
70 FORMAT( /,5X, 'THE EXPECTED CUSTOMER POPULATION AT STN.', I2, ' = Q(', I2, ', F12.6)
PRINT160, NS
160 FORMAT( /,5X, 'NUMBER OF JOBS IN THE SYSTEM FOR THIS EXAMPLE IS =', I13)
PRINT75, TQW
75 FORMAT( /,5X, 'THE TOTAL MEAN WAITING TIME FOR AN ARBITRARY JOB IS=', F12.6)
PO=R1+R2
PRINT76, PO
76 FORMAT( /,5X, 'THE PROCESSING POWER OF THE SYSTEM=', F12.6)
PRINT77, ZETA, ZET1
77 FORMAT( /,5X, 'ARRIVAL RATE AT THE CENTRAL STATION=', F12.6, /,
15X, 'ARRIVAL RATE AT THE MACHINING CELLS =', F12.6)
PRINT150, RO,R1,R2
150 FORMAT( ,5X,'AVERAGE UTILISATION OF SIN0 = ', F12.6,
1   ,5X,'AVERAGE UTILISATION OF SIN1 = ', F12.6,
1   ,5X,'AVERAGE UTILISATION OF SIN2 = ', F12.6)
STOP
END

//GO.SYSIN DD *
..... DATA CARDS ....

//
B.2 SAMPLE PROGRAM FOR THE APPROXIMATE MODEL

// JOBNAME JOB (XXX,XXX), 'D. GUPTA'; CLASS=Z
// FORIVCLG EXEC FORIVCLG
// FORT SYSIN DD *
C **********************************************************************************
C THIS PROGRAM CALCULATES THE STEADY STATE PROBABILITIES *
C AND SEVERAL PERFORMANCE MEASURES USING THE APPROXIMATE ANALYSIS *
C - METHOD # 01. *
C **********************************************************************************
DIMENSION A(50,50), B(50), X(25), WA(25), PI(25), WK(1300), C(4)
1, Q(3)
INTEGER IWK(25)
REAL LD, MT, MI, MII
C **********************************************************************************
C THE INPUT PARAMETERS ARE DEFINED AS FOLLOWS: *
C A = THE M BY N COEFFICIENT MATRIX. *
C B = THE M BY NB MATRIX OF RIGHT HAND SIDES. *
C C = COMMUNICATION VECTOR OF LENGTH 4 (INPUT IF IND=1, AND OUTPUT) *
C M = THE NUMBER OF ROWS IN A AND B. *
C N = THE NUMBER OF COLUMNS IN A AND ROWS IN X. (INPUT) *
C NB = NUMBER OF RIGHT HAND SIDES (COLUMNS IN B). (INPUT) *
C IA = ROW DIMENSION OF MATRIX A AS SPECIFIED IN THE CALLING PROG *
C IB = ROW DIMENSION OF MATRIX B AS SPECIFIED IN THE CALLING PROG *
C IX = ROW DIMENSION OF MATRIX X AS SPECIFIED IN THE CALLING PROG *
C IND = OPTION SELECTION INDICATOR (DEFAULT=0) *
C NS = 5 = THE NUMBER OF JOBS IN THE SYSTEM. *
C K = THE NUMBER OF AGVS. *
C IZ = LOCAL STORAGE SPACES AVAILABLE AT EACH MACHINING CELL. *
C LD = ARRIVAL DELAY AT THE CENTRAL STATION *
C MT = 0.5 = THE RATE AT WHICH JOBS ARE TRANSFERRED BY THE AGVS *
C MI = 0.2 = THE MACHINING RATE AT CELL # 1 *
C MII = 0.2 = THE MACHINING RATE AT CELL #2 *
C **********************************************************************************
IA=50
IB=50
READ(5,1) M,N,NB,IND,IX,NS,K,IZ,LD,MT,MI,MII
1 FORMAT(8I3,5X,4F6.2)
READ(5,2) C(1), C(2), C(3)
2 FORMAT(3F10.2)
DO 5 I=1,M
DO 5 J=1,N
A(I,J)=0.0
IF(I.EQ.M) A(I,J)=1.0
IF(I.EQ.J) A(I,J)=-1.0
CONTINUE

AVM=(MI+MII)/2
A(2,1)=1.0
A(3,2)=EXP(-(NS-1)*LD/MT)
A(4,2)=1-EXP(-(NS-1)*LD/MT)
A(1,3)=AVM/(AVM+(NS-1)*LD)
A(5,3)=((NS-1)*LD)/(AVM+(NS-1)*LD)
A(5,4)=1.0
A(2,5)=AVM*(1-EXP(-(AVM+(NS-2)*LD)/MT))/((NS-2)*LD+AVM)
A(9,5)=((NS-2)*LD*(1-EXP(-(AVM+(NS-2)*LD)/MT)))/((NS-2)*LD+AVM)
A(6,5)=(1/3)*EXP(-(AVM+(NS-2)*LD)/MT)
A(7,5)=(1/3)*EXP(-(AVM+(NS-2)*LD)/MT)
A(8,5)=(1/3)*EXP(-(AVM+(NS-2)*LD)/MT)
A(3,6)=(MI+MII)/(MI+MII)+(NS-2)*LD
A(10,6)=(NS-2)*LD/(MI+MII)+(NS-2)*LD
A(3,7)=MI/(MI+(NS-2)*LD)
A(11,7)=(NS-2)*LD/(MI+(NS-2)*LD)
A(3,8)=MI/(MI+(NS-2)*LD)
A(12,8)=(NS-2)*LD/(MI+NS-2)*LD
A(4,9)=1-EXP(-AVM/(2*MT))
A(10,9)=(1/3)*EXP(-AVM/(2*MT))
A(11,9)=(1/3)*EXP(-AVM/(2*MT))
A(12,9)=(1/3)*EXP(-AVM/(2*MT))
A(5,10)=((MI+MII)*(1-EXP(-(MI+MII)+(NS-3)*LD)/MT)))/((AVM+(NS-3)*LD)
A(13,10)=((NS-3)*LD*(1-EXP(-(MI+MII)+(NS-3)*LD)/MT)))/(AVM+(NS-13)*LD)
A(14,10)=(1/2)*EXP(-(MI+MII)+(NS-3)*LD)/MT)
A(16,10)=(1/2)*EXP(-(MI+MII)+(NS-3)*LD)/MT)
A(5,11)=(MT*(1-EXP(-(MI+NS-3)*LD)/MT))/((MI+(NS-3)*LD)
A(15,11)=((NS-3)*LD*(1-EXP(-(MI+(NS-3)*LD)/MT)))/(MI+(NS-3)*LD)
A(14,11)=EXP(-(MI+(NS-3)*LD)/MT)
A(5,12)=(MI+(1-EXP(-(MI+(NS-3)*LD)/MT)))/((MI+(NS-3)*LD)
A(17,12)=(NS-3)*LD*(1-EXP(-(MI+(NS-3)*LD)/MT)))/(MI+(NS-3)*LD)
A(16,12)=EXP(-(MI+(NS-3)*LD)/MT)
A(9,13)=1-EXP(-(MI+MII)/(2*MT))
A(18,13)=(1/2)*EXP(-(MI+MII)/(2*MT))
A(19,13)=(1/2)*EXP(-(MI+MII)/(2*MT))
A(6,14)=MI/((MI+MII+(NS-3)*LD)
A(7,14)=MII/((MI+MII+(NS-3)*LD)
A(18,14)=(NS-3)*LD/((MI+MII+(NS-3)*LD)
A(9,15)=1-EXP(-MI/(2*MT))
A(18,15)=EXP(-MI/(2*MT))
A(6,16)=MII/((MI+MII+(NS-3)*LD)
A(8,16)=MI/((MI+MII+(NS-3)*LD)
A(19,16)=(NS-3)*LD/((MI+MII+(NS-3)*LD)
A(9,17)=1-EXP(-MI/(2*MT))
A(19,17)=EXP(-MI/(2*MT))
A(11,18)=(MII*1-EXP(-(MI+MII)/MT))/(MI+MII)
A(10,18)=(MI*1-EXP(-(MI+MII)/MT))/(MI+MII)
A(20,18)=EXP(-(MT+MII)/MT)
A(10,19)=(MII*1-EXP(-(MI+MII)/MT))/(MI+MII)
A(12,19)=(MI*1-EXP(-(MI+MII)/MT))/(MI+MII)
A(20,19)=EXP(-(MT+MII)/MT)
A(14,20)=MI/((MI+MII)
A(10,20)=MI/((MI+MII)
PRINT10
10 FORMAT('1', 'THE FOLLOWING MATRIX REPRESEN'TS THE TRANSITION RATE MA
11 TRIX: ')
12 DO 15 J=1,N
13 DO 15 I=1,M
14 PRINT20 I,J,A(I,J)
20 FORMAT(5X,'A(,,I2,,,,I2,) = ',F12.6)
15 CONTINUE
16 DO 25 I=1,M
17 B(I) = 0.0
25 CONTINUE
26 CONTINUE
B(M)=1.0
C**** CALCULATE THE STEADY STATE PROBABILITIES OF IMBEDDED MARKOV CHAIN
17 CALL LSEQF (A, IA, M, N, B, IB, NB, IND, C, X, IX, IWK, WK,IER)
18 DO 30 I=1,N
19 PRINT31 I,X(I)
31 FORMAT(5X,'THE OUTPUT PROBABILITIES OF IMBEDDED MARKOV CHAIN ARE
32 1= X(,,I2,,') = ',F10.6)
30 CONTINUE
C**** MEAN RESIDENCY TIMES IN EACH STATE ARE CALCULATED AS FOLLOWS:
WA(1)= 1/(NS*LD)
WA(2)= (1-EXP(-(NS-1)*LD/MT))/((NS-1)*LD)
WA(3)= 1/(AVM+(NS-1)*LD)
WA(4)= 1/(2*MT)
WA(5)=1-EXP(-(AVM+(NS-2)*LD)/MT))/(AVM+(NS-2)*LD)
WA(6) = 1/(MII+MII+(NS-2)*LD)
WA(7) = 1/(MII+(NS-2)*LD)
WA(8) = 1/(MII+(NS-2)*LD)
WA(9) = (1-EXP(-AVM/(2*MII)))/AVM
WA(10) = (1-EXP(-(MII+MII+(NS-3)*LD)/MT))/(MII+MII+(NS-3)*LD)
WA(11) = (1-EXP(-(MII+MII+(NS-3)*LD)/MT))/(MII+(NS-3)*LD)
WA(12) = (1-EXP(-(MII+(NS-3)*LD)/MT))/(MII+(NS-3)*LD)
WA(13) = (1-EXP(-(MII+MII)/2MT))/(MII+MII)
WA(14) = 1/(MII+MII+(NS-3)*LD)
WA(15) = (1-EXP(-MII/2MT))/MII
WA(16) = 1/(MII+MII+(NS-3)*LD)
WA(17) = (1-EXP(-MII/2MT))/MII
WA(18) = (1-EXP(-(MII+MII)/MT))/(MII+MII)
WA(19) = (1-EXP(-(MII+MII)/MT))/(MII+MII)
WA(20) = 1/(MII+MII)

C**** CALCULATE THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV
C QUEUEING PROCESS.
SUM = 0.0
DO 40 I=1,N
SUM = SUM + WA(I)*X(I)
40 CONTINUE
DO 50 II=1,N
PI(II) = WA(II)*X(II)/SUM
50 CONTINUE
PRINT55
55 FORMAT('1',2X,'THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV P
1RCESS ARE:')
DO 65 I=1,N
PRINT60,I,I,PI(I)
60 FORMAT('0',5X,'PROB. THAT THE SYSTEM STATE IS "',I2,'" =',
1'PI(',I2,') = ',F12.6)
65 CONTINUE
C *******************************************************
C THE FOLLOWING SEGMENT OF THE PROGRAMME IS TO CALCULATE THE
C PERFORMANCE MEASURES:
C Q(I) = EXPECTED CUSTOMER POPULATION AT STATION I.
C TND = EXPECTED SYSTEM DELAY FOR AN ARBITRARY CUSTOMER
C ZETA = THRUPT PUT RATE OF THE COMPUTERISED MANUF SYSTEM
C R1 = AVERAGE UTILISATION OF M/CING CELL 1
C R2 = AVERAGE UTILISATION OF M/CING CELL 2
C *******************************************************
PRINT200, NS
200 FORMAT(/,5X,'THE NUMBER OF JOBS FOR THIS EXAMPLE IS=' ,I2)
Q(1) = NS*(PI(1)+PI(2)+PI(4)) + (NS-1)*(PI(3)+PI(5)+PI(9))
1+(NS-2)*PI(6)+PI(7)+PI(8)+PI(10)+PI(11)+PI(12)+PI(15)+PI(17)+PI(19)
1+2*(PI(7)+PI(11)+PI(15)+PI(19)+16)+PI(20)

Q(2) = 0.5*(PI(3)+PI(5)+PI(9)) + PI(6)+PI(10)+PI(13)+PI(16)+PI(19)
1+2*(PI(7)+PI(11)+PI(15)+PI(19)+16)+PI(20)

Q(3) = 0.5*(PI(3)+PI(5)+PI(9)) + PI(6)+PI(10)+PI(13)+PI(14)+PI(18)
1+2*(PI(8)+PI(12)+PI(16)+PI(17)+PI(19)+PI(20))

RL = PI(6)+PI(7)+PI(10)+PI(13)+PI(14)+PI(15)+PI(16)
1+2*(PI(7)+PI(11)+PI(15)+PI(19)+16)+PI(20)

R2 = PI(6)+PI(8)+PI(12)+PI(10)+PI(13)+PI(14)+PI(16)+PI(17)
1+2*(PI(18)+PI(19)+PI(20)+0.5*(PI(3)+PI(5)+PI(9))

ZETA = RL*MII + R2*MII

TW0 = (Q(1)+((Q(2)+Q(3))/2)+1)/ZETA

TW1 = (Q(1)+Q(2)+Q(3))/ZETA

TW2 = Q(1)/ZETA

TW3 = TW2 + 0.5*(Q(2)+Q(3))/ZETA

DO 66 I = 1, 3

IFL = I-1

PRINT70, IFL, IFL, Q(I)

CONTINUE

66 FORMAT(/,5X, 'THE EXPECTED CUSTOMER POPULATION AT STN.', I2, ' = Q(', I2, ')

PRINT75, TW0, TW1, TW2, TW3

70 FORMAT(/,5X, ' THE TOTAL MEAN WAITING TIME FOR AN ARBITRARY JOB IS:'

1./, ' TW0 = ', F12.6, ' TW1 = ', F12.6, ' TW2 = ', F12.6, ' TW3 = ', F12.6)

PO = RL+R2

PRINT76, PO

75 FORMAT(/,5X, ' THE TOTAL MEAN WAITING TIME FOR AN ARBITRARY JOB IS:'

1./, ' TW0 = ', F12.6, ' TW1 = ', F12.6, ' TW2 = ', F12.6, ' TW3 = ', F12.6)

PRINT77, ZETA

76 FORMAT(/,5X, ' THE PROCESSING POWER OF THE SYSTEM = ', F12.6)

PRINT78, ZETA

77 FORMAT(/,5X, ' THE THROUGHPUT RATE OF THE SYSTEM = ', F12.6)

PRINT150, R1, R2

150 FORMAT(/,5X, ' AVERAGE UTILISATION OF STN#1 = ', F12.6,

1, 5X, ' AVERAGE UTILISATION OF STN#2 = ', F12.6)

STOP

END
B.3 SAMPLE PROGRAM FOR THE APPROXIMATE MODEL#2

//JOBNAME JOB (XXXX,XXX), D. GUPTA', CLASS=Z
//FORTVCLG EXEC FORTVCLG
//FORT.SYSIN DD *
C ****************************************************************************************** *
C THIS PROGRAMME CALCULATES THE STEADY STATE PROBABILITIES *
C AND SEVERAL PERFORMANCE MEASURES USING THE APPROXIMATE ANALYSIS *
C METHOD # 02. *
C ****************************************************************************************** *
C DIMENSION A(50,50), B(50), X(25), WA(25), PI(25), WK(1300), C(4) * 
C 1, Q(3) *
C INTEGER IMK(25) *
C REAL LD, MT, MI, MII *
C ****************************************************************************************** *
C THE INPUT PARAMETERS ARE DEFINED AS FOLLOWS: *
C A = THE M BY N-COEFFICIENT MATRIX. *
C B = THE M BY NB MATRIX OF RIGHT HAND SIDES. *
C C = COMMUNICATION VECTOR OF LENGTH 4 (INPUT IF IND=1, AND OUTPUT) *
C M = THE NUMBER OF ROWS IN A AND B. *
C N = THE NUMBER OF COLUMNS IN A AND ROWS IN X. (INPUT) *
C NB = NUMBER OF RIGHT HAND SIDES (COLUMNS IN B). (INPUT) *
C IA = ROW DIMENSION OF MATRIX A AS SPECIFIED IN THE CALLING PROG *
C IB = ROW DIMENSION OF MATRIX B AS SPECIFIED IN THE CALLING PROG *
C IX = ROW DIMENSION OF MATRIX X AS SPECIFIED IN THE CALLING PROG *
C IND = OPTION SELECTION INDICATOR (DEFAULT=0) *
C NS = 5 = THE NUMBER OF JOBS IN THE SYSTEM. *
C K = THE NUMBER OF AGVS. *
C IZ = LOCAL STORAGE SPACES AVAILABLE AT EACH MACHINING CELL. *
C LD = ARRIVAL DELAY AT THE CENTRAL STATION *
C MT = 0.5 = THE RATE AT WHICH JOBS ARE TRANSFERRED BY THE AGVS *
C MI = 0.2 = THE MACHINING RATE AT CELL # 1 *
C MII = 0.2 = THE MACHINING RATE AT CELL #2 *
C ****************************************************************************************** *
C IA=50 *
C IB=50 *
C READ(5,1) M, N, NB, IND, IX, NS, K, IZ, LD, MT, MI, MII *
C FORMAT(BI3,5X,4F6.2) *
C READ(5,2) C(1), C(2), C(3) *
C FORMAT(3F10.2) *
C DO 5 I=1, M
DO 5 J=1,N
A(I,J)=0.0
IF(I.EQ.M) A(I,J)=1.0
IF(I.EQ.J) A(I,J)=-1.0
CONTINUE

AVM=(MI+MII)/2
A(2,1)=1.0
A(3,2)=1-EXP(-(NS-1)*LD/MT)
A(4,2)=EXP(-(NS-1)*LD/MT)
A(5,3)=1.0
A(1,4)=AVM/(AVM+(NS-1)*LD)
A(5,4)=(NS-1)*LD/(AVM+(NS-1)*LD)
A(2,5)=AVM*(1-EXP(-(AVM+(NS-2)*LD)/MT))/((NS-2)*LD+AVM)
A(6,5)=((NS-2)*LD*(1-EXP(-(AVM+(NS-2)*LD)/MT)))/((NS-2)*LD

A(7,5)=(1/2)*EXP(-(AVM+(NS-2)*LD)/MT)
A(8,5)=(1/2)*EXP(-(AVM+(NS-2)*LD)/MT)
A(3,6)=1-EXP(-AVM/(2*MT))
A(9,6)=0.5*EXP(-AVM/(2*MT))
A(4,7)=(MI+MII)/(MI+MII+(NS-2)*LD)
A(9,7)=(NS-2)*LD/(MI+MII+(NS-2)*LD)
A(4,8)=AVM/(AVM+(NS-2)*LD)
A(10,8)=(NS-2)*LD/(AVM+(NS-2)*LD)
A(5,9)=(2*AVM*(1-EXP(-(2*AVM+(NS-3)*LD)/MT)))/((2*AVM+(NS-3)*LD)
A(11,9)=((NS-3)*LD*(1-EXP(-(2*AVM+(NS-3)*LD)/MT)))/((2*AVM+(NS-3)*

A(12,9)=EXP(-(2*AVM+(NS-3)*LD)/MT)
A(5,10)=(AVM*(1-EXP(-(AVM+(NS-3)*LD)/MT)))/(AVM+(NS-3)*LD)
A(13,10)=((NS-3)*LD*(1-EXP(-(AVM+(NS-3)*LD)/MT)))/(AVM+(NS-3)*LD)
A(12,10)=EXP(-AVM+(NS-3)*LD)/MT)
A(6,11)=1-EXP(-(MI+MII)/(2*MT))
A(14,11)=EXP(-(MI+MII)/(2*MT))
A(7,12)=AVM/(MI+MII+(NS-3)*LD)
A(8,12)=AVM/(MI+MII+(NS-3)*LD)
A(14,12)=(NS-3)*LD/(MI+MII+(NS-3)*LD)
A(6,13)=1-EXP(-AVM/(2*MT))
A(14,13)=EXP(-AVM/(2*MT))
A(9,14)=(AVM*(1-EXP(-(MI+MII)/MT)))/(MI+MII)
A(10,14)=(AVM*(1-EXP(-(MI+MII)/MT)))/(MI+MII)
A(15,14)=EXP(-(MI+MII)/MT)
A(12,15)=1.0
PRINT10
10 FORMAT('1', 'THE FOLLOWING MATRIX REPRESENTS THE TRANSITION RATE MA
LTRIX: ') 
DO 15 J=1,N 
DO 15 I=1,M 
PRINT20, I, J, A(I, J) 
20 FORMAT(5X, 'A(12,12) = ', F12.6) 
15 CONTINUE 
DO 25 I=1,M 
B(I) = 0.0 
25 CONTINUE 
B(M) = 1.0 
C**** CALCULATE THE STEADY STATE PROBABILITIES OF IMBEDDED MARKOV CHAIN 
CALL LLBQF (A, IA, M, N, B, IB, NB, IND, C, IX, IWK, WK, IER) 
DO 30 I=1,N 
PRINT31, I, X(I) 
30 CONTINUE 
FORMAT(5X, ', THE OUTPUT PROBABILITIES OF IMBEDDED MARKOV CHAIN ARE 
1= X('','12,'') = ', F10.6) 
30 CONTINUE 
C**** MEAN RESIDENCY TIMES IN EACH STATE ARE CALCULATED AS FOLLOWS: 
WA(1) = 1/(NS*LD) 
WA(2) = (1-EXP(-(NS-1)*LD/MT))/((NS-1)*LD) 
WA(3) = 1/(2*MT) 
WA(4) = 1/(AVM+(NS-1)*LD) 
WA(5) = (1-EXP(-(AVM+(NS-2)*LD)/MT))/(AVM+(NS-2)*LD) 
WA(6) = (1-EXP(-(AVM+(2*MT)))/AVM 
WA(7) = 1/(MI+MII+(NS-2)*LD) 
WA(8) = 1/(AVM+(NS-2)*LD) 
WA(9) = (1-EXP(2*AVM+(NS-3)*LD)/MT))/(2*AVM+(NS-3)*LD) 
WA(10) = (1-EXP(-(AVM+(NS-3)*LD)/MT))/(AVM+(NS-3)*LD) 
WA(11) = (1-EXP(-(MI+MII)/(2*MT)))/(MI+MII) 
WA(12) = 1/(MT+MII+(NS-3)*LD) 
WA(13) = (1-EXP-(AVM/(2*MT)))/AVM 
WA(14) = (1-EXP(-(MI+MII)/MT))/(MI+MII) 
WA(15) = 1/(MI+ MII) 
C**** CALCULATE THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV C 
QUEUEING PROCESS. 
SUM = 0.0 
DO 40 I=1,N 
SUM = SUM + WA(I)*X(I) 
40 CONTINUE 
DO 50 II=1,N 
PI(II) = WA(II)*X(II)/SUM 
50 CONTINUE
PRINT55
FORMAT('l',2X,'THE STEADY STATE PROBABILITIES OF THE SEMI-MARKOV PROCESSES ARE:')
DO 65 I=1,N
PRINT60,I,1,PI(I)
60 FORMAT('0',5X,'PROB. THAT THE SYSTEM STATE IS ''',I2,''' =',
   1''PI(',I2,') = ',F12.6)
CONTINUE
C *******************************************************************************************
C THE FOLLOWING SEGMENT OF THE PROGRAMME IS TO CALCULATE THE PERFORMANCE MEASURES; 
C Q(I) = EXPECTED CUSTOMER POPULATION AT STATION I. 
C TWQ = EXPECTED SYSTEM DELAY FOR AN ARBITRARY CUSTOMER 
C ZETA = THRUPUT RATE OF THE COMPUTERISED MANUF SYSTEM 
C RL = AVERAGE UTILISATION OF M/CING CELL 1 
C R2 = AVERAGE UTILISATION OF M/CING CELL 2 
C *******************************************************************************************

PRINT200,NS
200 FORMAT(/,5X,'THE NUMBER OF JOBS FOR THIS EXAMPLE IS=',I2)
       Q(1)=NS*(PI(1)+PI(2)+PI(3))+(NS-1)*(PI(4)+PI(5)+PI(6))
       RL+(NS-2)*(PI(9)+PI(7)+PI(8)+PI(10)+PI(11)+PI(13))
       1+(NS-3)*(PI(14)+PI(12))+(NS-4)*PI(15)
       Q(2)= 0.5*(PI(4)+PI(5)+PI(6)) + PI(7)+PI(9)+PI(11)+PI(8)+PI(10)
       1+PI(13)+1.5*(PI(12)+PI(14)) + 2*PI(15)
       Q(3)= Q(2)
       RL= PI(7)+PI(9)+PI(11)+PI(12)+PI(14)+PI(15)
       1 + 0.5*(PI(4)+PI(5)+PI(6)+PI(8)+PI(10)+PI(13))
       R2= RL
       ZETA =R1*M1 + R2*MII
       TWQ=(Q(1)+((Q(2)+Q(3))/2)+1)/ZETA
       TW1=(Q(1)+Q(2)+Q(3))/ZETA
       TW2=Q(1)/ZETA
       TW3=TW2+0.5*(Q(2)+Q(3))/ZETA
       DO 66 I=1,3
       IFI=I-1
       PRINT70,IFI,IFI,PI(I)
66 CONTINUE
70 FORMAT(/,5X,'THE EXPECTED CUSTOMER POPULATION AT STN. ',I2,' = Q(',
       1'I2,')=',F12.6)
PRINT75, TWQ, TW1, TW2, TW3
75 FORMAT( '/ , 5X, 'THE TOTAL MEAN WAITING TIME FOR AN ARBITRARY JOB IS:'
1, '/ , 'TWQ=' , F12.6, ',', 'TW1=' , F12.6, ',', 'TW2=' , F12.6, ',', 'TW3=' , F12.6)
PO= R1 + R2
PRINT76, PO
76 FORMAT( '/ , 5X, 'THE PROCESSING POWER OF THE SYSTEM=' , F12.6)
PRINT77, ZETA
77 FORMAT( '/ , 5X, 'THE THRUPUT RATE OF THE SYSTEM=' , F12.6)
PRINT150, R1, R2
150 FORMAT( '/ , 5X, 'AVERAGE UTILISATION OF STN#1 = ', F12.6,
1 , 5X, 'AVERAGE UTILISATION OF STN#2 = ', F12.6)
STOP
END

//GO.SYSIN DD *

.... DATA CARDS ....

//
REFERENCES


VITA AUCTORIS

1960 Born in New Delhi, India on the 16th of October.

1977 Completed Higher Secondary education from Bal Bharati Public School, New Delhi, securing sixth rank in the order of merit of the All India Higher Secondary Examination.

1982 Graduated from the Indian Institute of Technology, New Delhi with a Bachelor of Technology Degree in Mechanical Engineering, securing first division with distinction.

1984 Currently a candidate for the M.A.Sc. degree in Industrial Engineering at the University of Windsor, Windsor, Canada.