

INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

- 1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.**
- 2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.**
- 3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.**
- 4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.**
- 5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.**

Xerox University Microfilms

300 North Zeeb Road
Ann Arbor, Michigan 48106

73-21,283

BERNARD, Prosper M., 1941-
USE OF COMPUTER SIMULATION TO TEST THE VALIDITY AND
THE SENSITIVITY OF REAL-TIME AND TIME-SHARING
QUEUEING MODELS.

The City University of New York, Ph.D., 1973
Business Administration

University Microfilms, A XEROX Company, Ann Arbor, Michigan

© COPYRIGHT BY
PROSPER M. BERNARD
1973

USE OF COMPUTER SIMULATION TO TEST THE VALIDITY AND THE
SENSITIVITY OF REAL-TIME AND TIME-SHARING QUEUING MODELS

by

PROSPER BERNARD

A dissertation submitted to the Graduate
Faculty in Business in partial fulfillment
of the requirements for the degree of Doctor
of Philosophy, The City University of New York.

1973

This manuscript has been read and accepted for the Graduate Faculty in Business in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

May 14, 1973
date

Lloyd Rosenberg
Chairman of Examining Committee

May 14, 1973
date

Conrad Berenson
Executive Officer

Prof. Herbert Arkin

Prof. N. Paul Loomba

Prof. Leon Winer

Prof. N. Hauser

Supervisory Committee

The City University of New York

ABSTRACT

USE OF COMPUTER SIMULATION TO TEST THE VALIDITY AND THE
SENSITIVITY OF REAL-TIME AND TIME-SHARING QUEUING MODELS

BY

Prosper Bernard

ADVISER: Professor Lloyd H. Rosenberg

The purpose of the research reported on in this dissertation was to test by simulation the analytical solution of some real-time and time-sharing queuing models and to test the sensitivity of those models to a change in the arrival distribution.

The models studied include a simple real-time model, a model for direct access, a time-sharing model with different priority disciplines such as Round-Robin, Feedback with infinite number of queues, shortest job first, First come First Served, a time-sharing with a limited number of queues, and a model for external priority.

In all cases a GPSS computer program is written and used to make experiments on the model. Confidence limits on the means are built by making many runs with different random numbers and therefore avoiding the autocorrelation inherent in pseudo-random numbers generation techniques.

The same programs are run with a departure from the classical "Poisson Arrival". Uniform arrival, constant arrival, and normal arrival are used. A two way analysis of variance with fixed factors is made to determine the effect of the arrival distribution, the level of utilization or another queue discipline and the interaction between the two factors.

The results indicate that many of the models do not behave everywhere as prescribed by mathematical formulae. Furthermore, it is shown that for certain models the arrival distribution does not have an important effect on the response time and for others it does.

ACKNOWLEDGEMENTS

It is a great pleasure to be able to acknowledge the help which I have received from many people while preparing this dissertation.

First, I wish to thank Prof. Lloyd H. Rosenberg, the chairman of my examining committee, Prof. Herbert Arkin, Prof. N. Paul Loomba, and Prof. Leon Winer, for their discerning criticisms, their invaluable suggestions, and their cooperation.

I wish to pay a general tribute to all professors of the doctoral program in Business especially to Dr. C. Berenson executive officer.. Next, I wish to thank the following organizations for providing computer facilities: The Bernard M. Baruch College, The Graduate division of the City University of New York. St. John's University, Maritime Overseas Corporation, The Provincial Bank of Canada, and The University of Montreal. Although many people have helped, I am the only one responsible for any errors and omissions.

Next, I would like to express my gratitude to my late parents who taught me, early in life, the great value of knowledge. And last, but far from the least, I wish to thank my wife, Francine, for her patience in preparing many drafts of this dissertation from my illegible handwriting while at the same time caring for our children. She has been cheerful, understanding, and always encouraging during the long and demanding process of doctoral studies. As she well knows, without her, I never would have achieved a goal which, in spite of my limitations, I had set up high in life. Although I have spent much time with my books and away from her, she is now and forever my only love.

Prosper M. Bernard
Bromont, Quebec
January 1973

TABLE OF CONTENTS

	PAGE
ABSTRACT	IV
ACKNOWLEDGEMENTS	V
TABLE OF CONTENTS	VI
LIST OF FIGURES	IX
LIST OF TABLES	XII
CHAPTER ONE - Introduction	I
Real-Time	2
Time-Sharing	3
Direct-Access	7
Configuration as Interaction of Systems	8
Review of The Literature	10
Contribution To Knowledge	15
CHAPTER TWO - Objective and Procedure	16
CHAPTER THREE - Computer Simulation Techniques	19
CHAPTER FOUR - A Real-Time Model	28
Model Description	29
Analytical Solution	33
Hypotheses	40
Simulation Results	40

CHAPTER FIVE - Models of Direct Access Method	55
Model Description	58
Analytical Solution	62
Hypotheses	69
Simulation Results	69
CHAPTER SIX - Time-Sharing Models	90
Models To Be Studied	93
Procedure	94
Round-Robin Scheduling	98
Feedback Model	124
Shortest Job First	146
CHAPTER SEVEN - Finite Time-Sharing Models	168
Finite Number of Customers	170
Model With Finite Queues	190
CHAPTER EIGHT - Model With Priority	205
Model To Be Studied	207
Analytical Solution	211
Hypotheses	215
Simulation Results	215
CONCLUSION	226

APPENDIX ONE - Program RA10V1 Random Access One Channel	234
APPENDIX TWO - Program TS50 - Test of Time-Sharing	237
APPENDIX THREE - Program RL10 - Real-Time Model	252
APPENDIX FOUR - Program CONF1M - Confidence Limits	256
APPENDIX FIVE - Program FINAUAUV - Analysis of Variance	257
APPENDIX SIX - Configuration of IBM 2311	262
APPENDIX SEVEN - Program FINCON01 - Confidence Limits	263
APPENDIX EIGHT - Program PR10 - Priority Model	268
APPENDIX NINE - Program TS70 - Time-Sharing With Many Queues	277
APPENDIX TEN - Proof of Waiting Time Theorem -RR Model	284
APPENDIX ELEVEN - Proof of Waiting Time Theorem-FB Model	288
APPENDIX TWELVE - Proof of Waiting Time Theorem-SJF Model	291
APPENDIX THIRTEEN - Proof of Response Time Theorem - FBN Model	298
APPENDIX FOURTEEN - Proof of Response Time Theorem - FBN Model	304
APPENDIX FIFTEEN - Response Time Theorem For External Priority	310
APPENDIX SIXTEEN - Program FIM105	318
BIBLIOGRAPHY	320
AUTOBIOGRAPHY	333

LIST OF FIGURES

	PAGE
1.1 COST OF RESPONSIVENESS FOR ON-LINE COMPUTER SYSTEMS	6
1.2 BASIC FUNCTIONS OF REAL-TIME SYSTEMS	9
4.1.1 SIMPLE REAL-TIME INQUIRY MODEL	29
4.1.2 SIMPLE REAL-TIME INQUIRY MODEL	30
4.1.3 SIMPLE REAL-TIME INQUIRY MODEL	31
4.1.4 MODEL OF MACHINE INTERFERENCE	32
4.1.5 POISSON RATIO FUNCTION	34
4.1.6 RESPONSE TIME VERSUS INQUIRY RATE:ANALYTICAL SOLUTION	39
4.2.1 RESPONSE TIME VERSUS INQUIRY RATE:SIMULATION RESULTS	49
5.1.0 ACCESS MECHANISM	56
5.1.1 DIRECT ACCESS MODEL	60
5.1.2 ACCESSES PER SECOND VERSUS NUMBER OF DISKS	67
5.1.3 INTER-ARRIVAL TIME VERSUS THE NUMBER OF DISKS	68
5.2.1 ACCESSES PER SECOND FOR PERFECT SERVICE	71
5.2.2 INTER-ARRIVAL TIME FOR PERFECT SERVICE	72
5.2.3 WAITING TIME WITH CONFIDENCE LIMITS	77
5.2.4 WAITING TIME SUMMARY	80
5.2.5 RESPONSE TIME FOR POISSON ARRIVAL	82
5.2.6 RESPONSE TIME FOR UNIFORM ARRIVAL	83
5.2.7 RESPONSE TIME FOR NORMAL ARRIVAL	84
6.1.1 TIME-SHARING MODELS	95
6.1.2 TIME-SHARING MODELS WITH N QUEUES	96
6.1.3 COMPARISON OF FB AND RR WAITING TIME	103
6.2.1 WAITING TIME VERSUS UTILIZATION-ALL MODELS	106
6.2.2 WAITING TIME VERSUS UTILIZATION-ALL MODELS	107
6.2.3 WAITING TIME VERSUS UTILIZATION-FSFC MODEL	108
6.2.4 WAITING TIME UNIT WITH LESS THAN ONE QUANTUM	118
6.3.1 FB MODEL	126
6.3.2 EXPECTED WAITING TIME	130
6.4.1 WAITING TIME PRELIMINARY RESULT	133
6.4.2 WAITING TIME VERSUS UTILIZATION-FB MODEL	139
6.4.3 WAITING TIME FOR THE FB? MODEL	140
6.5.1 WAITING TIME - SJF MODEL	149
6.6.1 WAITING TIME - SJF MODEL	151

6.6.2	WAITING TIME VERSUS UTILIZATION - PR MODEL	153
6.7.1	COMPARISON OF DISCIPLINES	157
6.7.2	WAITING TIME: SJF - FCFS	159
6.8.1	WAITING TIME - FB MODEL	161
6.8.2	WAITING TIME - FB MODEL	162
6.8.3	WAITING TIME - FB, FCFS, SJF MODELS	163
6.8.4	WAITING TIME - RR, FB MODELS	164
6.8.5	WAITING TIME - SJF MODEL	165
7.1.1	MODEL OF TIME-SHARING	167
7.1.2	EXPECTED TOTAL RESPONSE TIME VERSUS QUANTUM SIZE	175
7.1.3	EXPECTED TOTAL RESPONSE TIME VERSUS QUANTUM SIZE	175
7.1.4	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	176
7.1.5	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	176
7.1.6	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	176
7.1.7	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	176
7.2.1	EXPECTED TOTAL RESPONSE TIME VERSUS QUANTUM SIZE	184
7.2.2	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	185
7.2.3	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	186
7.2.4	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	187
7.2.5	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF service	188
7.3.1	FB MODEL	189
7.3.2	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	195
7.3.3	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	196
7.3.4	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	197
7.3.5	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	197

7.4.1.	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	201
7.4.2	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	202
7.4.3	RESPONSE TIME AS A FUNCTION OF THE LENGTH OF SERVICE	203

LIST OF TABLES

	PAGE
4.1.1 REAL-TIME MODEL - ANALYTICAL RESULTS	38
4.2.1 EXAMPLE OF ACTUAL SIMULATION OUTPUT OF PROGRAM RL10	43
4.2.2 EXAMPLE OF ACTUAL SIMULATION OUTPUT OF PROGRAM RL10	44
4.2.3 RESPONSE TIME - REAL-TIME	45
4.2.4 ANALYTICAL RESULTS VERSUS SIMULATION RESULTS	46
4.2.5 CONFIDENCE LIMITS FOR RESPONSE TIME	48
4.2.6 SAMPLE OUTPUT OF CONFIDENCE FINDING PROGRAM CONLMI	50
4.2.7 ANALYSIS OF VARIANCE: EFFECT OF ARRIVAL DISTRIBUTION	55
5.1.1 ARRIVAL FOR NEAR PERFECT SERVICE	66
5.2.1 SAMPLE OUTPUT OF THE SIMULATION PROGRAM RAIOVI	73
5.2.2 SAMPLE OUTPUT OF THE SIMULATION PROGRAM RAIOVI	74
5.2.3 SUMMARY OF TOTAL WAITING TIME	75
5.2.4 CONFIDENCE LIMITS FOR WAITING TIME	76
5.2.5 SUMMARY OF RESPONSE TIME	79
5.2.6 ANALYSIS OF VARIANCE: EFFECT OF ARRIVAL PATTERN	85
5.2.7 ANALYSIS OF VARIANCE: EFFECT OF ARRIVAL PATTERN	86
5.2.8 INFERENCE FROM A SAMPLE	87
6.2.1 WAITING TIME FOR DIFFERENT DISCIPLINES	105
6.2.2 SAMPLE OUTPUT OF THE SIMULATION PROGRAM TS50	109
6.2.3 SAMPLE OUTPUT OF THE SIMULATION PROGRAM TS50	110
6.2.4 SAMPLE OUTPUT OF THE SIMULATION PROGRAM TS50	111
6.2.5 SUMMARY OF WAITING TIME IN THE QUEUE - RR MODEL	113
6.2.6 CONFIDENCE LIMITS FOR WAITING TIME -	114

6.2.7	CONFIDENCE LIMITS FOR WAITING TIME - RR7	114
6.2.8	CONFIDENCE LIMITS FOR WAITING TIME - RR3 MODEL	114
6.2.9	ONE SIDED TEST RR MODEL	115
6.2.10	RESPONSE TIME	120
6.2.11	RESPONSE TIME	121
6.2.12	ANALYSIS OF VARIANCE-EFFECT OF THE ARRIVAL PATTERN - RR MODEL	122
6.2.13	ANALYSIS OF VARIANCE-EFFECT OF THE ARRIVAL PATTERN-RR MODEL	127
6.4.1	WAITING TIME ALL DISCIPLINES	132
6.4.2	WAITING TIME FB MODEL	135
6.4.3	CONFIDENCE LIMITS FOR WAITING TIME - FB MODEL	136
6.4.4	CONFIDENCE LIMITS FOR WAITING TIME - FB3 MODEL	136
6.4.5	CONFIDENCE LIMITS FOR WAITING TIME -FB7 MODEL	137
6.4.6	CONFIDENCE LIMITS FOR THE UTILIZATION	137
6.4.7	RESPONSE TIME FOR THE UNITS REQUIRING MORE THAN ONE QUANTUM	142
6.4.8	RESPONSE TIME FOR ALL UNITS	143
6.4.9	ANALYSIS OF VARIANCE - EFFECT OF THE ARRIVAL PATTERN - FB MODEL	144
6.4.10	ANALYSIS OF VARIANCE - EFFECT OF THE ARRIVAL PATTERN - FB MODEL	145
6.6.1	SUMMARY OF RESPONSE TIME - SJF MODEL	154
6.6.2	SUMMARY OF WAITING TIME - SJF MODEL	154
6.6.3	SUMMARY OF WAITING TIME - SJF MODEL	154
6.6.4	ANALYSIS OF VARIANCE - SJF MODEL	155
6.6.5	ANALYSIS OF VARIANCE - SJF MODEL	156
7.2.1	SUMMARY OF RESPONSE TIME-MANY CUSTOMERS	179
7.4.1	SUMMARY OF RESPONSE TIME-MANY QUEUES	199

8.2.1	WAITING TIME IN A PRIORITY MODEL	216
8.2.2	CONFIDENCE LIMITS ON WAITING TIME	217
8.2.3	SAMPLE OUTPUT OF THE SIMULATION PROGRAM PR10	218
8.2.4	SAMPLE OUTPUT OF THE SIMULATION PROGRAM PR10	219
8.2.5	RESPONSE TIME IN A PRIORITY MODEL	220
8.2.6	RESPONSE TIME IN A PRIORITY MODEL	223
8.2.7	ANALYSIS OF VARIANCE-EFFECT OF THE ARRIVAL DISTRIBUTION	224

CHAPTER ONE

INTRODUCTION

In order to understand the purpose of this dissertation and its contribution to the advancement of knowledge, it is important to describe the computer systems which are highly dependent on queuing processes, and to review the work that has been done in dealing with these processes. It will then be possible to show how the research reported in this dissertation will add to the present knowledge.

Since the work of A.K. Erlang¹, on automatic telephone exchanges, many scholars have been successful in applying Queuing Theory to other fields².

¹ E. Brockmeyer, H.L. Halstrom, and A. Jensen. "The Life and Works of A.K. Erlang", Translations of the Danish Academy of Technical Sciences, No.2, 1-277 (1948)

² The progress of Queuing Theory is primarily due to the following publications: E.C. Molina. "Application of the Theory of Probability to Telephone Trunking Problems", Bell System Technical Journal, Vol. 6 (1927), pp. 461 ff; T.C. Fry, "Probability and Its Engineering Use", (Princeton, N. J.: D. Van Nostrand Company, Inc.), 1928; A. Kolmogorov. "Sur le Problème d'Attente", Matematicheskii Sbornik, Vol. 38 (1931), pp. 101 ff; C.D. Crommelin. "Delay Probability Formulae When the Holdines are Constant", Post Office Electrical Engineering Journal, Vol. 25 (1932); pp. 55 ff; A. Khintchine. "Standard Poisson-Input Arbitrary-Holding Single-Channel Problems", Matematicheskii Sbornik, Vol. 30 (1932), pp. 73 ff; R. Kronig. "On Time Losses in Machinery Undergoing Interruption Physica", Vol. 10 (1943), pp. 215 ff; E.G. Bowen and T. Percy. "Delay in Air Traffic Control", Journal of The Royal Aeronautical Society, (1948), pp.251-258; D.G. Kendall. "Some Problems in the Theory of Queues", Journal of the Royal Statistical Society, Series B, Vol. 13, No. 2, (1951); J. Riordan. "Telephone Traffic Time-Average", Bell Systems Technical Journal, Vol. 30 (1951); and "Delay Curves for Calls Served at Random", Bell Systems Technical Journal, Vol. 32, No. 1 (1953).

difficult to establish a time criterion to determine whether or not a system is real-time. The processing must be done in a sufficiently rapid manner to influence the process being controlled³. For example, in a real-time credit system the credit officer wants to access the file and control the credit before the inventory leaves the premises but without delay in normal customer service. Waiting lines are then formed at each location for the transmission line and at the unique central processing unit for access to the processing. Queuing lines are also formed in accessing a mass storage device where the information is kept.

TIME-SHARING

Since they require real-time systems, time-sharing systems appeared only after these; at the same time they require the sharing of the CPU by two programs. Time-sharing was the next step in the development of computer technology. Computers were becoming bigger and bigger but still were used for small programs; The CPU was used only a fraction of the time. The logical step in the development of computer systems was then to share the large core among many programs in such a way that there would be little excess capacity, short waiting lines to the computer, and little waste of CPU time.

In a time-sharing system each user is given a "quantum" of exclusive service. At the end of this short period of time the processing is transferred to another user according to a scheduling plan or formula. If a user does not complete his work in the quantum he goes back and waits at the end of the same queue or at the end of another waiting line.

³ A formal definition of all the terms presented in this chapter can be found in Charles Sippl, "Computer Dictionary and Handbook", (New York: Howard W. Sam & Co., INC., (1966), and in Martin H. Weild, "Dictionary of Computer Terms", (New York: Hayden), 1969.

The Queuing Theorists have found applications for most processes having random arrivals with limited service, but their efforts have been seriously hampered by complex mathematical expressions which are difficult to solve. The advent of the computer permitted a broadening of the scope of the theory in two ways. First, it became easier to solve complex mathematical expressions by the use of algorithms written to quickly handle such problems, while simulation techniques were used to solve empirical problems. Second, the fast expansion of the computer industry and the advancement of its technology has created interesting applications in the past few years on which very little has been published.

In this thesis the computer will be considered both as a subject and a tool. On the one hand, a computer system will be studied in itself; on the other hand, the computer will be treated as an instrument to solve a previously analyzed problem. Therefore the computer systems which lead to complex queuing problems will now be introduced and the models and solutions presented in the literature will be reviewed. In the next chapter we will see how the computer can be used effectively to solve the queuing problems.

Real-Time. A real-time system must meet two basic requirements. First, it must be capable of maintaining a current global data base. The information must be stored on file and updated to reflect changes. Secondly, the system must be able to bring to the decision maker the necessary information at the proper time and place. Usually, the computer is accessed via common carrier lines from different locations.

The central processor must respond to inputs within a very short time, perhaps a few seconds. However, it is

Since the communication with the computer is done via common carrier lines at a very slow speed (in comparison with the CPU speed) it is possible for the CPU to do a lot of processing while information goes back to the users. A time-sharing system usually has five characteristics⁴:

1. On-Line: The equipment or terminals are under the control of the CPU and any user has the ability to interact with the CPU.
2. Real-Time: The computer must be able to service all users on a real-time basis within the requirements of the individual applications being processed. A problem will be solved sufficiently quickly to give an answer within the actual time limitations inherent in the problem. Another dimension of real-time systems was discussed at the beginning of the chapter.
3. Concurrent Access: Two or more users must be able to interact with the computer from remote locations concurrently.
4. Integrity: Each user must be able to use the computer as if it were exclusively dedicated to him. The records of one user cannot be modified or used by another unless planned in this way.
5. Functional Latitude: The system can be designated to a single application for specific users or can serve different types of users.

Models of Time-Sharing. There are three main models for time-sharing:

1. Round-Robin model.
2. Foreground-Background model, also called Feedback.
3. External Priority model.

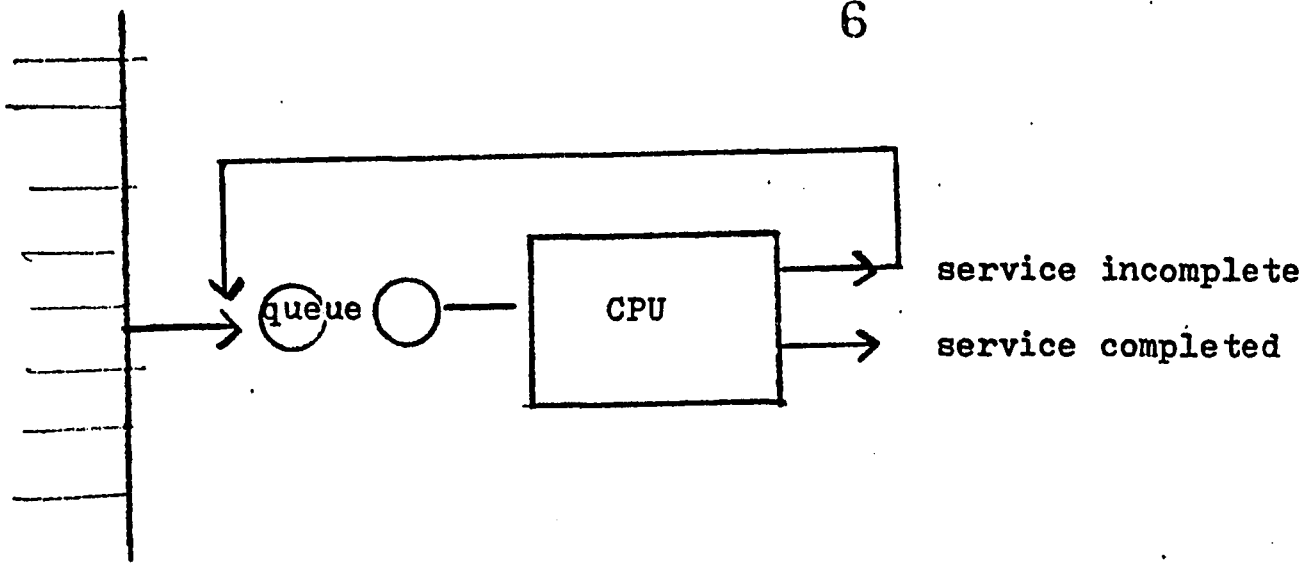
⁴ James Ziegler, Time-Sharing Data Processing Systems, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967), p. 15.

In the Round-Robin model shown in figure 1.1, arrivals for service come and form one queue. Each user is assigned a maximum quantum of service in the system. If a unit has received service within that time, it goes out; if not, it goes to the end of the queue.

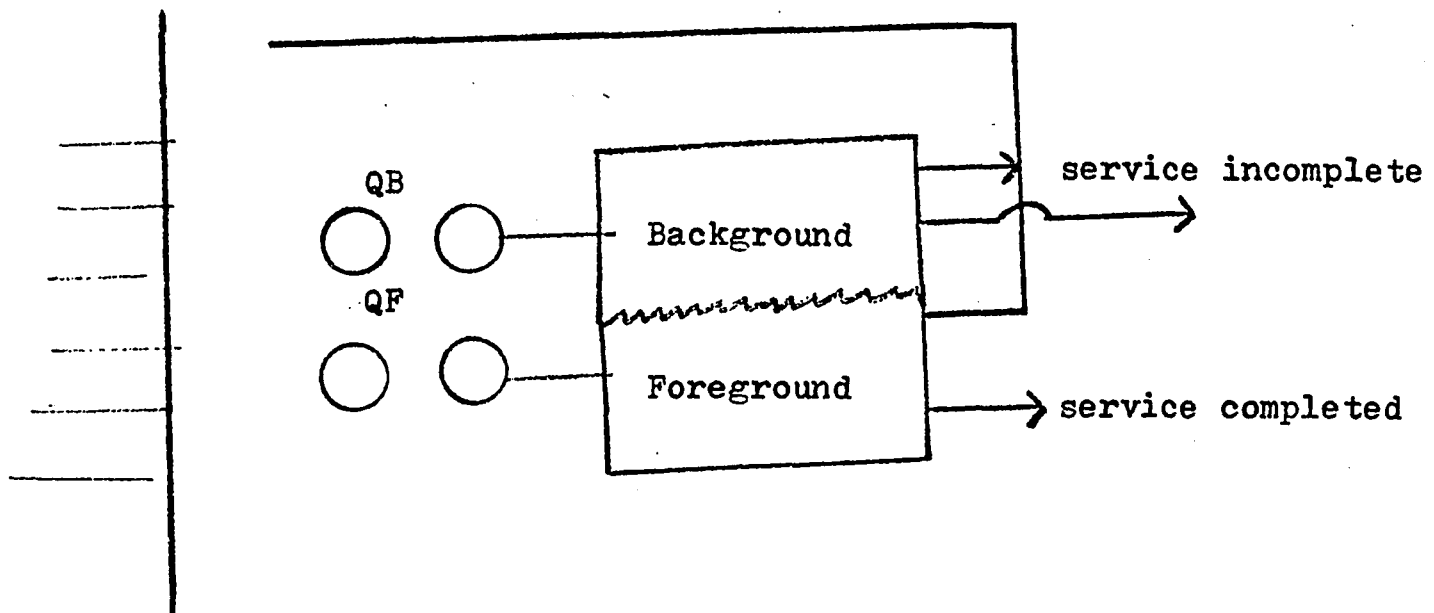
In the foreground-background model, each unit is assigned a maximum quantum of service in the foreground. If service is not completed within that time, the unit goes to the end of the background queue. The second queue is serviced only when a foreground I/O operation is needed or when there are no more units to serve in the foreground.

One can also construct a model where priorities for service are assigned by an external mechanism or according to a formula.

The time-sharing system can be complex. There can be many programs or users in core at a time, each partition having a quantum of service. When service exceeds the quantum in one partition (which is usually the case) the program is put on a direct access storage device. There is a maximum number of such transfers in and out of core to and from the disk or drum.



ROUND-ROBIN MODEL



FOREGROUND-BACKGROUND MODEL

Figure 1.1 - Two Models for Time-Sharing

Source: J. M. Mekinney. "A Survey of Analytical Time-Sharing Models", Computing Survey, Vol. 1, No. 2, (June 1969), p. 106.

DIRECT ACCESS

In any time-sharing or real-time system there is not only a transfer of information from the central processing unit to the user but also a transfer of information (program, routine, or data) from the CPU to a direct access or random access device.

One of the basic capabilities of a real-time system is to maintain a current global base. This data must be accessed rapidly enough to meet the response time requirements. This is made possible by storing information on a device from which it can be obtained randomly. In other words, the position from which the information is to be obtained is in no way dependent on the position from which previous information was obtained.

In sequential access storage devices such as tapes or cards, the only way to access a particular record is to read all preceding records on the file. This set-up is certainly impractical for a real-time reservation system. Since the customer is waiting for an answer within a few seconds, it is inefficient to go through all the files to find the proper flight. The system must be able to associate with the flight number a certain location on the disk and directly access that location.

The accessing of a mass storage device is slow compared to the CPU speed. The position of the record must be found on the device and the information transmitted via a channel. There are usually a large number of direct access devices and a small number of channels on a computer system, and waiting lines are formed at a different stages.

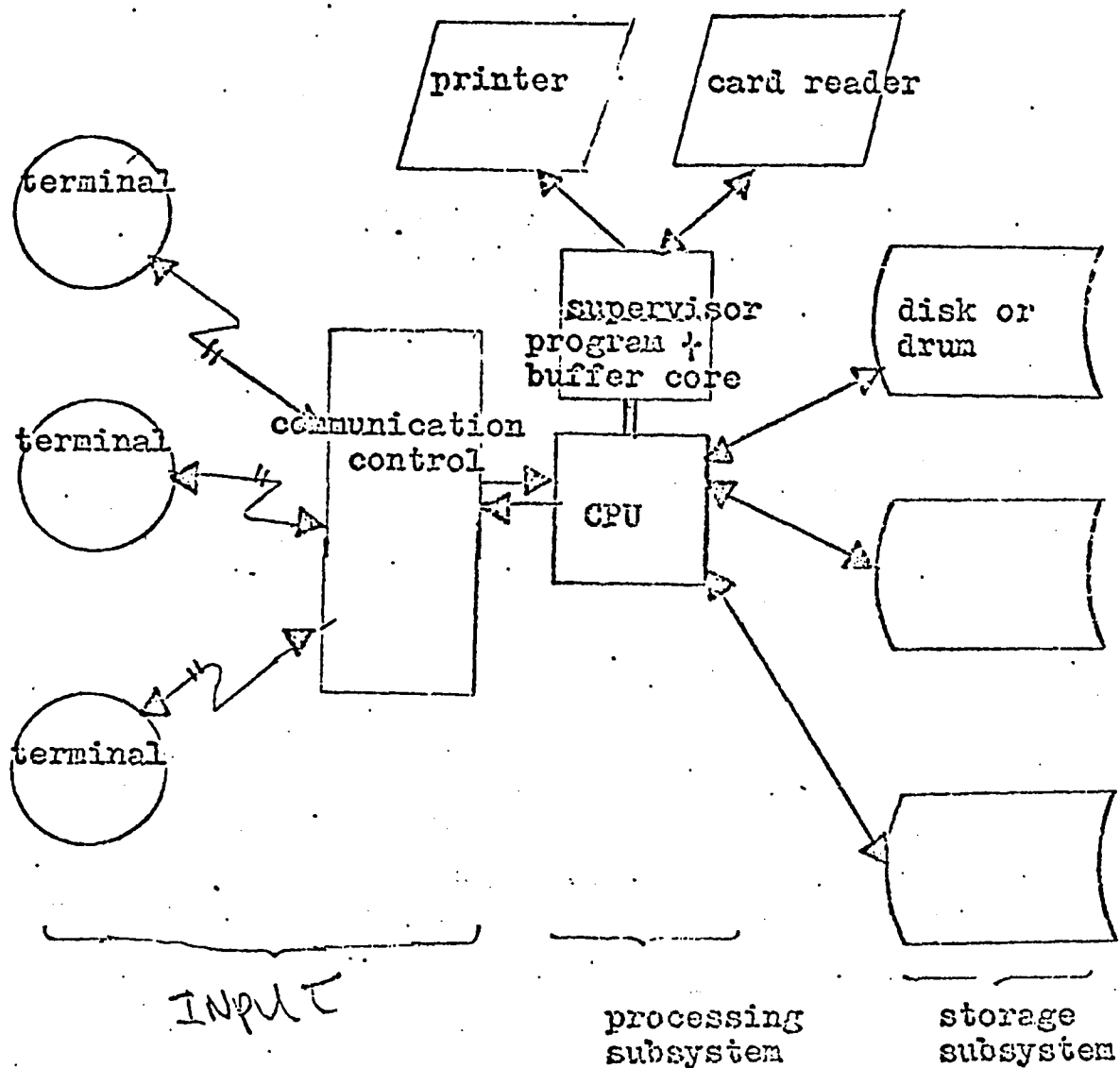
CONFIGURATION AS INTERACTION OF SYSTEMS

The main advantages of real-time and time-sharing systems are derived from the interaction of earlier functions of data processing associated with the technology of a telecommunication network. There are three basic functions of a real-time system: (1) the input, consisting of the terminal sub-system and the communication sub-system; (2) the processing; and (3) the random access of a file. It is the effective interaction of these three functions that determine the efficiency of the system.

As shown in figure 1.2, the system communicates with a man - a scientist, a programmer, or a clerk - via a terminal which may be either a typewriter or a television-like CRT and light-pen. The system may contain other I/O units for other batch processing programs which use the computer at the same time as the real-time system. In a teleprocessing system, I/O includes terminals and communication lines.

It is possible to process many users because the time required for information transmission over common carrier lines is longer than the processing time. Most third generation computers can be used for time-sharing or real-time without modification of the CPU. However, there are computers intended specifically for time-sharing.

The processing of either usual real-time inquiries or time-sharing accesses is controlled by an executive program or monitor which accepts inquiries and assigns priorities according to time of arrival or other factors. The executive program also buffers the information going to and from terminals and to and from direct access storage devices.



Input: a) terminal subsystem
 b) communication subsystem

Figure 1.2 - Basic functions of real-time and time-sharing systems.

Source: Developed by the author.

Since access to CPU is random and requires a real-time response, care must be taken both in the way in which the information is stored in a direct access storage device and in the way in which the information is transmitted.

The Possible Bottlenecks. If the time estimate for the normal batch processing is too low, the error can be compensated for by extra-shift working. However, it is not as easy to expand the real-time capacity because, as previously stated, it is the result of the interaction of sub-systems. There are many possible bottlenecks which can be caused by the terminal, the communication line, the central processing unit, the direct access devices, the channel for transmission of information from the mass storage devices, or even by the discipline by which the customers are serviced.

REVIEW OF THE LITERATURE

In the past few years, many papers have presented analytical solutions to different queuing models in computer systems. Most solutions presented are dependent on restrictive assumptions as it will be explained later.

Schrage⁵ and Chang⁶ developed and solved analytically a model of real-time with priorities when the arrival distribution for each of the priority classes is Poisson.

⁵ L. Schrage. "Analysis and Optimization of a Queuing Model of a Real-Time Computer Control System", IEEE Transactions on Computers, Vol. C-18, No. 11, (November 1969), pp. 997-1004.
L. Schrage. "Some Queuing Models for a Time-Shared Facility", Ph.D. Dissertation, Engineering, Cornell University, February 19

⁶ W. Chang. "Single Server Queuing Processes in Computing System" IBM Systems Journal, Vol. 9, No. 1, (January 1970), pp. 36-71.
W. Chang. "Computer Congestion Analysis. A Theoretical Treatment" Polytechnic Institute of Brooklyn, Ph.D. Dissertation, 1969.

Chang expresses the need to use a Poisson arrival process for ease of solution. A non-Poisson arrival implies a non-Markovian stochastic process. Many applications of queuing theory to the solution of real-time model have been offered by IBM⁷ for many of the problems found in real-time systems. However, the application is limited to Poisson arrival streams and the analytical solutions of the models use approximation procedures.

Models and solutions have also been presented for accessing mass storage facilities as a sub-system of a larger system. Abate, Dubner, and Weinberg⁸ have found by analytical methods the response time, i.e. the time interval between the moment of arrival and departure in the system for a specific storage device with a fixed number of disk units. They assume that the rest of the system will generate to this sub-system a Poisson stream. Coffman⁹ extended the work of Denning and Weingarten¹⁰ to the queuing problems in a drum storage device. He obtained the queue length distribution and showed how this specific device requires a unique analysis. As before, the analysis applies only to a system where the arrival (or the output of the previous system) is a Poisson process.

⁷ IBM. "Analysis of Some Queuing Models in Real-Time Systems", (New York, IBM, Ref. F20-0007-1, 1968).

⁸ J. Abate, H. Dubner, and Weinberg. "Queuing Analysis of the IBM 2314 Disk Storage Facility", Journal of the ACM, Vol. 15 No. 4, (October 1968), pp. 577-589.

⁹ E.G. Coffman. "Analysis of a Drum Input/Output Queue", Journal of ACM, Vol. 16, No. 1, (1969) pp. 73-90.

¹⁰ Denning. "Effect of Scheduling on File Memory Operations", Proceeding AFIPS Spring Joint Computer Conference, Vol. 30, pp. 9-21. A. Weingarten. "The Eschenbach Drum Scheme", Communication of ACM, Vol. 9, No. 7, (July 1962), pp. 509-517.

Krishnamoorthi and Wood¹¹ found analytically the waiting time for a Round Robin model with a fixed set of inputs and a Poisson arrival process. Kleinrock¹² analyzed a Round Robin model without overhead and with Poisson arrival. Coffman and Kleinrock¹³ have expanded that study to different queue disciplines such as many levels of feedback and shortest queue first but with the same arrival distribution as the other ones. They found an expression for the expected waiting time and the expected waiting time conditional on the length of service required.

Rash¹⁴ includes the set-up time in his comparison of Round-Robin and Priority scheduling in terms of costs. He points out the need to use the Poisson Input Process to compare results with others but he indicates the needs to develop solutions for non-Poisson arrival processes.

¹¹B. Krishnamoorthi and R. L. Wood. "Time-Shared Computer Operations with Both Inter-arrival time and Service Time Exponential", Journal of ACM, Vol. 13, No. 3, (1966) pp 317-338

¹²L. Kleinrock. "Analysis of a Time-Shared Processor", Naval Research Logistics Quarterly 11, No. 10, (1964), pp. 59-73
L. Kleinrock. "Time-Shared Systems: a Theoretical Treatment", Journal of the ACM, Vol. 14, No. 2, (April 1967), pp. 242-261.

¹³E. Coffman and L. Kleinrock. "Feedback Queuing Models for Time-Shared Systems", Journal of the ACM, Vol 15, No. 4 (1968), pp. 549-576.

¹⁴P.J. Rash. "A Queuing Theory Study of Time-Shared Computer Systems", Ph.D. Dissertation, Department of Electrical Engineering, Southern Methodist University, August 1967.
P.J. Rash. "Queuing Theory Study of Round-Robin Scheduling of Time-Shared Computer System", Journal of ACM, Vol. 17, No. 1, (January 1970), pp. 131-145.

Shemer¹⁵ studied the same priority problems in a time-sharing environment with the same assumptions about the arrival process. He obtained an expression for the expected waiting time conditional on the length of service required. Chang¹⁶, expanding earlier work by Takacs¹⁷, arrived at the transient and stationary queue distribution for a time-sharing model assuming for ease of solution that there is a fixed probability of returning in the queue at the end of the quantum. The quantum, instead of being fixed, becomes a random variable related to the length of service required. It requires that each job be a multiple of the quantum.

Adiri and Avi-Itzhak¹⁸ have studied some time-sharing models where there is a limited number of feedback queues and a limited number of sources. They found that in each case the expected response time is conditional on the service requirement. In a finite source model, they used the machine interference model to deduce the expected response time and the response time conditional on the job size. Their solution is different from that presented by Krishnamoorthi and Wood.

-
- 15 J. Shemer. "Some Mathematical Considerations of Time-Sharing Scheduling Algorithms", Journal of ACM, Vol. 14, No. 7, (April 1967), pp. 262-272.
 - 16 W. Chang. "A Queuing Model for A Simple Case of Time-Sharing" IBM Systems Journal, Vol 5, No. , (1966), pp. 115-125
 - 17 L. Takacs. "A Single Server Queue with Feedback", The Bell System Technical Journal, pp. 505-519, (March 1963).
 - 18 I. Adiri and B. Avi-Itzhak. "A Time-Sharing Model with Many Queues", Operations Research, Vol. 17, No. 6, (November 1969), pp. 1077-1089. I. Adiri and B. Avi-Itzhak. "A Time-Sharing Queue", Management Science, Vol. 15, No. 11, (July 1969), pp. 639-657. I. Adiri and B. Avi-Itzhak. "A Time-Sharing Queue with a Finite Number of Customers", Journal of the ACM, Vol. 16, No. 2, (April 1969), pp. 315-323.

Recently, Heacox and Purdom¹⁹ have developed a modified Round-Robin with overhead into which the amount of service per pass depends on the arrival rate. The system is studied in terms of the mean cost due to the waiting time in queue. The results are somewhat different to the ones proposed by Rash, who also used the cost of delay as a criterion of efficiency.

Most of the models in the literature have in common an analytical treatment of queuing models with a strong requirement for a Poisson arrival process. No solution is offered for a departure from that assumption, although many authors have expressed the need to obtain a solution for non-Poisson arrival processes. Many models, as will be discussed in detail in later chapters, are solved by breaking down a system into sub-systems solving each of them independently, and adding up the solution. Assumptions are therefore made in between. In the second part of this dissertation, some of the models mentioned above will be tested. It is not possible to test all the models presented in the literature, but significant models in each of the inter-connected areas of real-time, time-sharing, and direct access will be studied. For example, in selecting time-sharing models for analysis, models have been included which offer the solution to several service procedures, queue disciplines, feedback procedures, and to other variable characteristics of time-sharing models²⁰.

¹⁹ H. Heacox and P.W. Purdom. "Analysis of Two Time-Sharing Queuing Models", Journal of ACM, Vol. 19, No. 1, (January 1972), pp. 70-91.

²⁰ As the reader will recall, a time-sharing system must also provide real-time.

CONTRIBUTION OF THIS DISSERTATION

This dissertation will advance knowledge in this area in three ways. First, we will test the validity of some of the solutions proposed in the literature. Second, we will test the sensibility of the models to the most critical assumption - the Poisson arrival. Third, we will present a systematic way to make experiments on the models through simulation. Unlike analytical techniques, the simulation sampling experiments developed in this dissertation will make it possible to study models with any type of arrival distribution. Simulation should also provide the decision-maker with an insight relevant to a range of parameter values. Sensitivity analysis can determine a range of distributions and parameters for which a set of insights are valid and hence reduces the requirement for costly field testing of the assumptions.

For the manager who has to make decisions under conditions of risk or partial ignorance, sensitivity analysis, as proposed in this dissertation, allows him to determine the responsiveness of the measures of the system to change in the parameter values or in the assumptions. Knowing in advance the sensitivity of the system to a change in the most critical assumptions, i.e. the arrival distribution to the system, the manager could determine in advance the need to purchase more information. If the system is sensitive to the arrival pattern and if it is found not to be Poisson, the manager cannot use the results of the analytical solution. However, simulation experiments can be made with any form of arrival distribution and therefore provide the decision-maker with complete information. Many of the simulation studies made in different fields often neglect to validate the results²¹. This dissertation will present also a systematic way to statistically validate the results of sampling experiments on the models.

²¹ In order to appreciate the need for systematic statistical analysis and experimental design in simulation one can review the papers presented at recent conferences on the application of simulation. For example, see Proceedings of the Fifth Conference on the Application of Simulation, (New York, (December 1971).

CHAPTER TWO

OBJECTIVE & PROCEDURE

The purpose of this dissertation is to test by simulation the validity of the analytical solution of some real-time and time-sharing queuing models and to test the sensitivity of those models to a departure from the classical "Poisson" assumption. This dissertation will also present a systematic way to use simulation as an experimental technique in studying complex models and in statistically validating the experimental results.

Why Test The Models? Most of the models studied involve a series of many stages. The usual proposed analytical solutions in the literature involve solving each stage independently and then adding up the results. This procedure implies that assumptions must be made at each stage, i.e. one must make assumptions on the output of a queuing process because each stage is a queuing process. It has always been difficult to obtain the distribution of the output of even a simple queuing process. It is therefore conceivable that the assumptions might be wrong for complex cases

The purpose of this dissertation is not to find the exact distribution at each stage but to study the behavior of the whole system. By simulating the same models, i.e. running sampling experiments on the models, one does not need to know the distributions at each stage because the output of stage $(n - 1)$ is used as input to stage (n) . There is no assumption made in between concerning the output of a queue. Therefore it is possible that by studying the system as a whole rather than a sum of stages, one may obtain different results.

The reader will notice that, as in any model, only some aspects of the real world are included. It is not the purpose of this dissertation to validate the models in terms of the real world but to test the validity of the analytical solutions which have been presented in the literature. However, the

author tests the effect of neglecting some aspects of the real world, by using sensitivity analysis on the assumptions.

Most of the queuing models studied in this dissertation assume a Poisson arrival distribution and no analytical solution is offered for any departure from that assumption. It might be easier to determine in advance whether the arrival distribution has any effect on the model, rather than to make a field study to make sure that the arrival is in fact Poisson. Even if one could determine the exact shape of the distribution, and it were found to be different from the Poisson distribution, no analytical solution is offered. Therefore, the following hypotheses will be tested:

- Hypotheses: A. The solution of the models is valid, that is the results of the analytical solution are statistically the same as the results of the simulation.
- B. The form of the arrival distribution does not have a significant effect on the response time, that is, the time in the system, and there is no interaction effect.

Models to be Studied. The models studied in this dissertation include a simple real-time model in chapter four, a model for direct access in chapter five, a time-sharing model with different priority disciplines such as Round-Robin, Feedback with infinite number of queues, Shortest Job First, and First Come, First Served in chapter six, a time-sharing with a limited number of queues in chapter seven, and a model for external priority in chapter eight. All the models studied are closely related to each other. In all cases, there is access to a single computer from different locations. Different procedures for servicing the customers lead to different models. In real-time or time-sharing systems an important sub-system is the access to a mass storage device which will be also studied

Testing Procedure. For all the models studied, a GPSS computer program is written and used to make experiments on the model. However, data generated by dynamic simulation experiments usually are highly autocorrelated not only because of the random number generation procedures but also because of the queuing processes themselves. If we assume the absence of autocorrelation we may underestimate the variance or we may take a too small sample. In order to avoid the autocorrelation effect, many independent runs or experiments are made. The means of all the runs are used as a sample. It is then possible to obtain a confidence interval around the mean of this sample. For each run the results of the transient period are disregarded and the model is studied in steady state.

The next step involves comparing the analytical results with the simulation results and finding out whether or not the analytical results fall within the 95% or 99% confidence interval. The first hypothesis is rejected if at any point the analytical results are outside the confidence intervals.

In order to test the second hypothesis, the same procedure is repeated for different distributions of the inter-arrival time. The author used distributions very different in nature such as uniform, normal, constant, and Poisson as described later in this dissertation. The mean of the inter-arrival time is the same in all cases. Having many values of the response time i.e. the time in the system for each arrival pattern in the system and for each level of utilization it is easy to perform a two-way analysis of variance. This procedure indicates whether either of the two factors have a significant effect and whether there is an interaction effect between them. The level of utilization, the second factor, has a significant effect and is used only to measure the interaction. It is possible, for example, that the arrival distribution has an effect only if the service station is loaded. This is why the interaction effect is also important.

The author has concentrated on testing the sensitivity of the system to the arrival distribution because, as shown in the literature review, this is most urgently needed.

CHAPTER THREE

COMPUTER SIMULATION

The purpose of this chapter is to describe the method that the author will use to test the hypotheses concerning some queuing models.

Nature of Simulation. The simulation technique does not try to isolate the relationship among the variables. It observes the way in which variables of the model vary over time. Gordon says that simulation is essentially an experimental problem-solving technique¹. Therefore, it is a technique for solving problems by following the changes over time of a dynamic model or system. Instead of trying to find the optimum solution of a model by mathematical techniques, the components are varied until the optimum solution has been found. We can define system simulation as experimental techniques to solve problems by following changes over time of a dynamic complex and usually stochastic system. It is then an inductive experiment involving sampling from a finite or an infinite population.

Monte Carlo Techniques. Monte Carlo techniques can be used to solve either deterministic or stochastic problems. Originally Monte Carlo denoted the introduction of random variables into the solution of a deterministic problem by setting up an isomorphism between this problem and some problems in probability theory which is then solved, for example, by performing sampling experiments². One can

¹ Gordon, "Systems Simulation", (Englewood Cliff, N.J. Prentice Hall, 1969), p. 10

² K.D. Tocker, "Simulation Languages", Progress on Operations Research, Vol. 111, (New York: Wiley, 1969), p. 75.

find the value of integrals using this method.

The term is also used to solve stochastic problems and this is the meaning it will have in this thesis. Monte Carlo involves statistical sampling³. Typically, simulated sampling involves replacing the actual universe of items by its theoretical counterpart, a universe described by some assumed probability distribution, and then sampling from this theoretical population by means of a random number table⁴. This technique can very easily be applied to queuing problems. By assigning an inter-arrival time to all numbers in the table, one can simulate the real life. In simulating queuing processes, one is not always interested in the average but in the peak. An airport, for example, must be able to handle its peak load rather than its average load.

A Monte Carlo estimate does not result in the true value but only in one possible value. Therefore, repeated sampling experiments are necessary to obtain a variance. There exist excellent publications concerning the use of random numbers in Monte Carlo simulation⁵.

Simulation by Computer. Although many types of simulation techniques have been proposed since the introduction of the Monte Carlo method, their use has not been widespread; taking a significant sample is time consuming and expensive. The characteristics of the computer, however, are ideally suited for use in simulation. First, the speed of the computer insures that simulation can be performed quickly. Second, its logical

³ Joseph A. Panico, Ibid, p. 130

⁴ Maurice Sasieni, Arthur Yospan, and Laurence Freedman, Operations Research Methods and Problems, (New York: John Wiley & Sons, Inc., 1959). p. 58

⁵ J. M. Hammersley, and D. C. Handscomb, Monte Carlo Methods, (New York: John Wiley & Sons, Inc., 1954); Y. Shreider, The Monte Carlo Method, (New York: Pergamon Press, 1965); J. Certaine "On Sequences of Pseudo-Random Numbers of Maximal Length," Journal of ACM, Vol. 5 (1958), pp. 353-356; and IBM, Random Number Generators

capability makes possible the testing of many alternatives. Third, the computer has a memory capable of keeping track of many different events going on at the same time.

To take advantage of such a computer, the user must construct a model representing the system and he must write a program to carry out the changes resulting from the sequence of events that affect the system. As well said by Gordon, to make a computer simulate a system, it is necessary to create an image of the system in the form of a set of numbers that represents the state of the system and to manipulate this image by writing a computer program that embodies the relationships controlling the changes of state in the system. The systems start at one state and combined effects of controllable and uncontrollable events cause the system to go to another state.

In order to simulate any system one must follow general steps:⁶

1. Definition of the problem
2. Planning the study
3. Formulation of the mathematical model
4. Construction of a computer program
5. Validation of the model
6. Design of experiments
7. Execution of simulation run and analysis of the results
8. Validation of results

⁶ G. Gordon, System Simulation, (Englewood Cliff, New Jersey: Prentice-Hall, 1969), p. 10

The first three steps are obvious and common to any O.R. problems. One has to decide on what aspects of the system performance are significant for the problem on hand. In this dissertation the "response time" is the measure of efficiency.

When the mathematical model has been constructed the process of writing a program is a well defined task as we shall see later.

The model must be validated as representing the aspect of the real world we are testing. The computer program must be validated as representing the mathematical model.

Computer Program. In order to properly simulate the states of any model on a computer, one must have the following:

1. A means of generating random numbers.
2. Some way of producing sample values from specified distributions.
3. Efficient programming techniques.

Various simulators have been written to meet the three requirements mentioned above.

Generation of Random Numbers. The basic to any Monte Carlo is a sequence of random numbers. Numbers can be obtained in two ways.

1. random numbers stored on disk or in memory from a table of true random numbers.
2. a set of pseudo-random numbers can be obtained by an arithmetical procedure.

Reading in core a table of one million true random numbers requires too much memory. For this reason most simulation programs generate their own sequence of pseudo-

random numbers. They are generated by a process that is deterministic but gives sequences that appear random and are independent of the rest of the simulation. There are few techniques available to generate such numbers.

Simulation Languages. The problem is now to translate the model constructed earlier into an image that can be written in one of several general purpose languages such as COBOL, FORTRAN, or PL/1. However, the wide use of computer simulation has led to the development of simulation languages.

Systems are classified as continuous or discrete, and the programming languages are usually designed for continuous or discrete systems. The different computer languages available for simulation are GPSS, SIMSCRIPT, DYNAMO, SIMULA, CSS, SIMPAC, CSL, GASP, FORSIM, and CSMP⁷. The two most commonly used languages for the simulation of discrete systems are GPSS (General Purpose System Simulator) and SIMSCRIPT. GPSS is a block diagram transaction-oriented language easy to learn and easy to use. However, by making it simple one loses some of the capabilities of a computer. On the other hand, SIMSCRIPT is a FORTRAN - like event oriented language, very powerful, but difficult to program and to debug.

Selection of a Language. There are two points to consider in selecting a simulation language.

1. Characteristics of the Language⁸ and of the problem including:

⁷ A good analysis of different simulation languages has been presented by Philip Kiviat in "Simulation Languages" in T. H. Nayfar, Computer Simulation of Experiments with Models of Economic Systems, (New York, John Wiley & Sons, 1971), pp. 406-

⁸ Sisson., op. cit. p. 146

- . time-advance method
 - . event orientation
 - . random number generation and variate generation capabilities
 - . state and entity variable
 - . output available and statistical analyses that can be performed
 - . capability to invert sub-routine.
2. The following points regarding the language itself must be considered:
- . availability of user's manual
 - . diagnostics
 - . easy to learn or not
 - . cost
 - . compiling and running time.

In this dissertation, the author will use GPSS which has been designed to solve queuing problems. GPSS is well supported and well described in many publications. GPSS is available on the University's computer⁹.

Validation of Results. One of the most important aspects of simulation is often in practice neglected¹⁰. One must statistically validate the results. A number of procedures have been suggested for dealing with problem of estimating the reliability of simulation results by

⁹ G. Gordon "A General Purpose Simulator", IBM System Journal (September 1962); GPSS/360 (poughkepsie, N.Y.: IBM ref. H20-0304, 1968; G. Gordon, op.cit., pp. 191-231; GPSS/360 Uses Manual, (poughkepsis, N.Y. IBM, ref. H20-0320, 1968).

¹⁰ In order to appreciate the need for systematic statistical analysis and experimental design in simulation one can review the papers presented at recent conferences on the application of simulation. For example, see, Proceeding of the Fifth Conference on the Application of Simulation, (New York, December 1971)

Naylor¹¹, Van Horne¹², Fishman and Kiviat¹³, Moy¹⁴, and Fishman¹⁵. In this dissertation we will make use of the technique described below.

Since we want to test the system at a steady state it is often useful and meaningful to load the system before keeping any results. For example, in simulating the queues formed at an airport it will not be meaningful to start the system assuming that all service facilities and queues are empty. The bias can be removed by loading initially known values into the system or by running the model without keeping results until it has reached the steady state level.

The system has reached steady-state conditions when successive observations of the system's performance are statistically the same. It can be measured when the number of observations in which the output is greater than the average at a point is statistically the same as the number in which it is less.

¹¹T. Naylor, K. Wertz, and T. Wonnacott, "Method of Analysing Data from Computer Simulation Experiments", Communication of ACM, Vol. 10, No. 11, (November 1967). _____, J. Kleijner and T. Seaks, "The Use of Multiple Ranking Procedures to Analyse Simulation of Management Systems: A Tutorial", Management Science, Vol. 18, No. 6, (February 1962) pp. B245-B257. _____, and I.M. Finger, "Verification of Computer Simulation Models", Management Science, Vol. 14, No. 2, (October 1967), pp. B92-B101. _____, et al. "The Application of Sequential Sampling to Simulation: An Example Inventory Model", Communication of the ACM, Vol. 13, No. 5, (May 1970), pp. 287-296. _____, Computer Simulation Experiments with Models of Economic Systems, (New York, John Wiley & Sons, 1971). _____, et al. "Spectral Analysis of Data Generated by Simulation Experiments with Economic Model", Econometrica (April 1969), pp. 333-352.

¹²Richard L. Van Horn, "Validation of Simulation Results", Management Science, Vol. 17, No. 15, (January 1971), pp. 247-258

¹³G. Fishman and P. Kiviat, "The Analysis of Simulation Generated Time Series", Management Science, Vol. 13, No. 7, (March 1967), pp. 525-557.

¹⁴W. Moy, "Variance Reduction", in Computer Simulation Experiments, T.H. Naylor (New York: John Wiley & Sons, 1971), pp. 269-287.

¹⁵George Fishman, "Estimating Sample Size on Computing Simulation Experiments", Management Science, Vol. 18, No. 1, (September 1971), pp. 21-39.

In establishing confidence intervals it is assumed that the observations are independent. Often values are autocorrelated. Under broad conditions that can normally be expected to hold in a simulation run, the sample means of autocorrelated data can be shown to approximate a normal distribution as the sample size n increases. However, the variance of autocorrelated data is not related to the population variance by the simple expression σ^2/n as it occurs in independent data. A term must be added to account for the autocorrelation. The term is positive in most cases and if it is ignored the confidence interval is excessively optimistic.

In this dissertation, several experiments (runs) will be done with the same sample size n in order to obtain a set of independent values of the means ($\bar{x}$). The batch means so obtained are independent and normally distributed according to the central limit theorem.

In this dissertation, some of the effects of such factors as the arrival distribution will be tested using the analysis of variance. The analysis of variance is a collection of techniques which are appropriate when the factors are qualitative like the type of arrival distribution. There are three basic assumptions:

1. Independence
2. Equality of variance
3. Normality

By taking the means of each run as input to the test we satisfy both the first and third assumptions provided we have enough means. Scheffé argues that inequality of variances has little effect if the cell numbers are equal ¹⁶.

CONCLUSION

This chapter has reviewed the basic principles of simulation and has described the way it will be used in the second part. The following chapters will then study specific and closely inter-connected queuing models in computer systems. For each of the models considered, simulation experiments will be made to test the validity of the solution and to test the sensitivity of the solution to a departure from the Poisson arrival. As pointed out by Van Horne¹⁷, simulation should provide the insight relevant for a range of parameter values. Sensitivity analysis can formally determine the set of distribution and parameters values for which a set of insight is relevant. We then reduce the requirement and the cost of empirically testing the assumptions.

¹⁶ Henry Scheffé, The Analysis of Variance, (New York, John Wiley & Sons, 1959), pp. 351-358.

¹⁷ Richard Van Horn, "Validation of Simulation Results", Management Science, Vol. 17, No. 5, (January 1971), p.251

CHAPTER FOUR

A REAL-TIME MODEL

The purpose of this chapter is to test by simulation the validity of the analytical solution of a simple real-time inquiry model and to test the sensitivity of that model to a departure from the classical "Poisson" arrival assumption. The method used is to describe the model which has been published and to discuss an analytical solution¹. As will be seen in detail below, the procedure for solving this multi-stage model of queues in series is first to solve each stage and then to add up the results. This implies making assumptions at each stage, and the procedure may introduce errors. For this reason, the simulation will be used to study the system as a whole without having to make intermediate assumptions. Independent simulation experiments will be made in order to obtain confidence limits on the mean response time.

Most queuing models, such as the one presented in this chapter, assume a Poisson arrival distribution and no solution is offered for a departure from the "Poisson" assumption. It might be easier to determine in advance the effect of the arrival distribution than to make field studies to make sure that it is in fact Poisson. Even if the exact shape of the distribution could be determined, and it were found to be different from the Poisson distribution, no solution is offered. For this reason the author will test the sensitivity of the model by repeating simulation experiments with different distributions such as Poisson, Normal, and Uniform.

¹ Analysis of Some Queuing Models in Real-Time Systems, (New York: IBM, Ref: F20-0007-1, 1966), p. 44-46.

MODEL TO BE TESTED

In this real-time inquiry model there are a finite number of customers linked to a central processor via common carrier lines and a terminal. A graphical representation of the model is given in figure 4.1.1 and with an example in figure 4.1.2

The following assumptions have been made:

1. The arrival distribution at each terminal is Poisson with mean λ/m where m is the number of terminals.
2. The service time of the central processor is distributed according to an Erlang-2 distribution.
3. The service times for each terminal i.e. the key-in time and print-out are uniformly distributed.
4. The queue discipline is First-in-First-out.

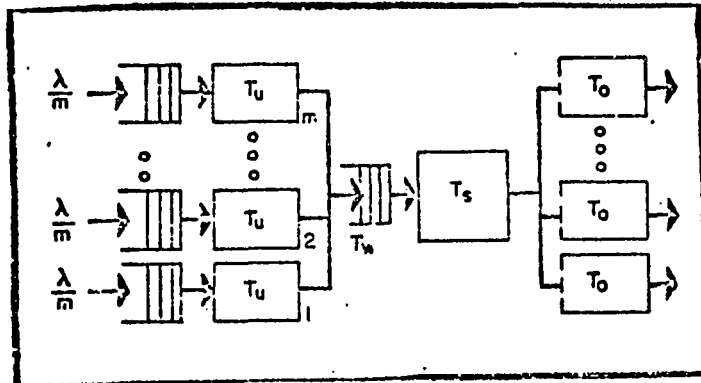


Figure 4.1.1 Simple Real-Time Inquiry Model

Source: Analysis of Some Queuing Models, (New York, IBM, Ref: F20-0007-1, 1968), p. 43

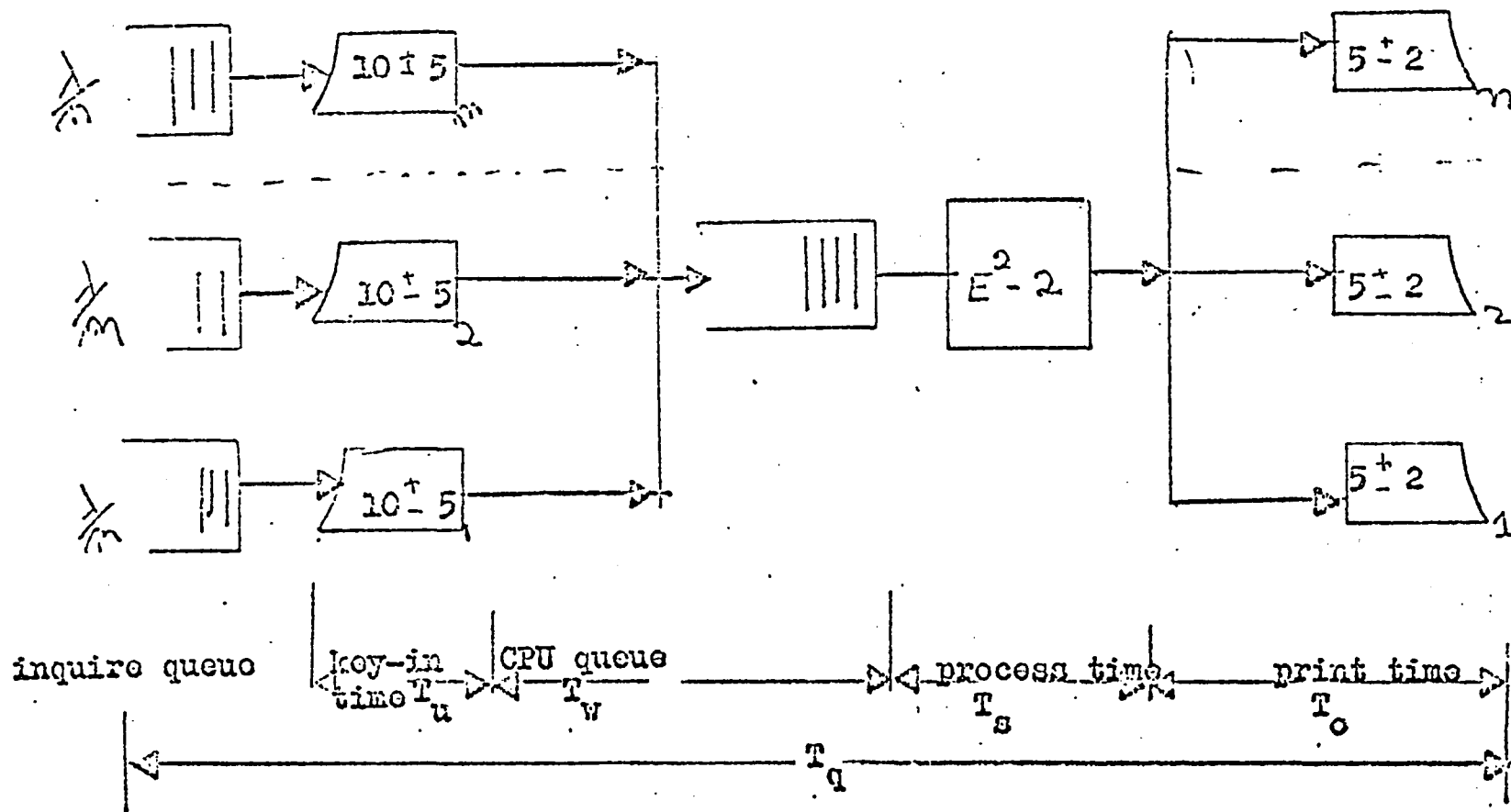


FIGURE : 4.1.2 : Single real-time inquiry model

Source: Modification of 'Analysis of Some Queuing Models' (New York, IBM ref F20-0007-1, 1963), p. 44.

The real-time inquiry model considered in this chapter could be seen as a machine interference model where there are external arrivals to each machine or, in this case, to each terminal. Customers arrive at each of the m terminals randomly with mean arrival rate λ/m . The total response time consists of the waiting-time for the terminal, the service-time of the terminal (key-in), the waiting for the CPU, the service-time in CPU, ^{and} the service-time of the terminal (Printing-out). The system is considered busy during the whole period and can accept a new customer only after the terminal processing of the previous customer.

The reader will ^{see} by looking at figures 4.1.3 and 4.1.4 the similarity between this model and the classical "machine interference" model. The whole model can be seen as a multi-stage model when the middle stage is a machine interference model. This latter model has been well described and analyzed in the literature.

In the machine interference model there are a finite number of machines or sources assigned to one service man or service station. The machine is either "up" or "down". When the machine goes down it joins the queue for service. The machine gets immediate service or wait for service depending on the availability of the single repair man. The machine interference model is illustrated in figure 4.1.4.

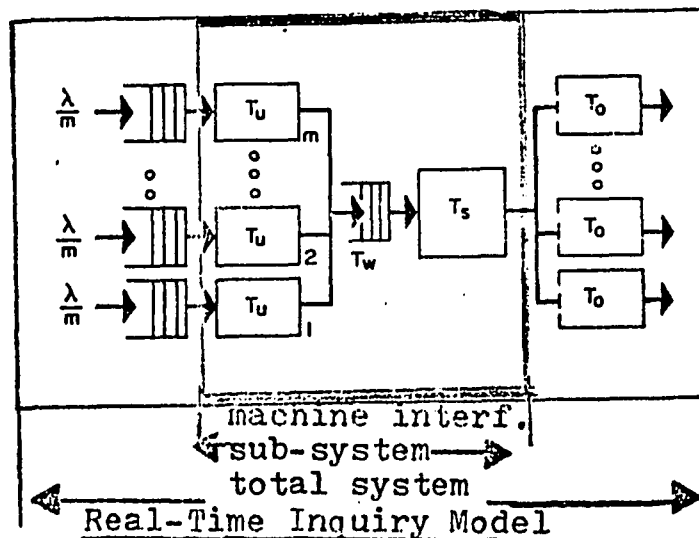


Figure 4.1.3

Real-Time Inquiry Model

Source: Modification of figure 4.1.1

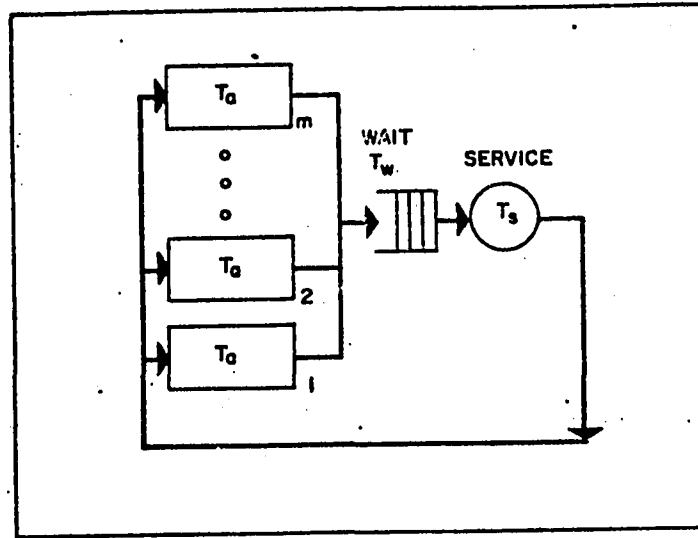


Figure 4.1.4 Machine Interference Model

Source: Analysis of Some Queuing Models, p. 39.

In the machine interference model the following assumptions are usually made:

1. The service time is exponentially distributed with mean $\bar{T}_s$.
2. The "up" time for each machine is exponentially distributed with mean time $\bar{T}_a$.

These assumptions are sometimes referred to as the worst-case conditions. The ratio of the two times is defined as the "service ratio" where

$$z = \frac{\bar{T}_a}{\bar{T}_s}$$

The "server Utilization" denoted as $r_m(z)$ can be obtained for each number of machines m and for each value of z as follows: Let the probability P_0 represent the fraction of time there are zero machine in the service queue, and the serviceman is idle. Thus, $1 - P_0$ may be interpreted as the fraction of time the serviceman is busy, that is, the server utilization:

$$\text{Server Utilization} = 1 - P_0 = 1 - \frac{e^{-z} \frac{z^m}{m!}}{e^{-z} \sum_{j=0}^m \frac{z^j}{j!}} = r_m(z) \quad 4.1.01$$

ANALYTICAL SOLUTION

The model has not been given a rigorous solution, but an approximation where the results of all stages are added up to make the total response time. The following solution has been proposed.²

From the user's point of view, the system is in use when he starts keying the inquiry so that it could be considered as a service-station or a "black box" in which service time T_p is:
 $T_p = T_u + T_w + T_s + T_o$ ³ 4.1.1

where

T_u = Time to key-in and transmit the message

T_w = Waiting time to access CPU

T_s = Time for service in the CPU

T_o = Time to transmit and print the result

T_p = Total service time

T_q = Total response time = waiting for the system + T_p

All the above times are assumed to be independent random variables. Therefore the variance of the total service time is the sum of variances.

$$\sigma_p^2 = \sigma_u^2 + \sigma_w^2 + \sigma_s^2 + \sigma_o^2 \quad 4.1.2$$

The model is represented in figure 4.1.2.

Since the overall system can be considered to be a single service station where the expected service-time is $\bar{T}_p$, the system utilization by a user can be defined as:

$$\rho = (\lambda/m) / (1/\bar{T}_p) = \frac{\lambda}{m} \cdot \bar{T}_p \quad 4.1.3$$

² Ibid

³ In general in this dissertation $\bar{x}$ represents the expected value of the random variable x .

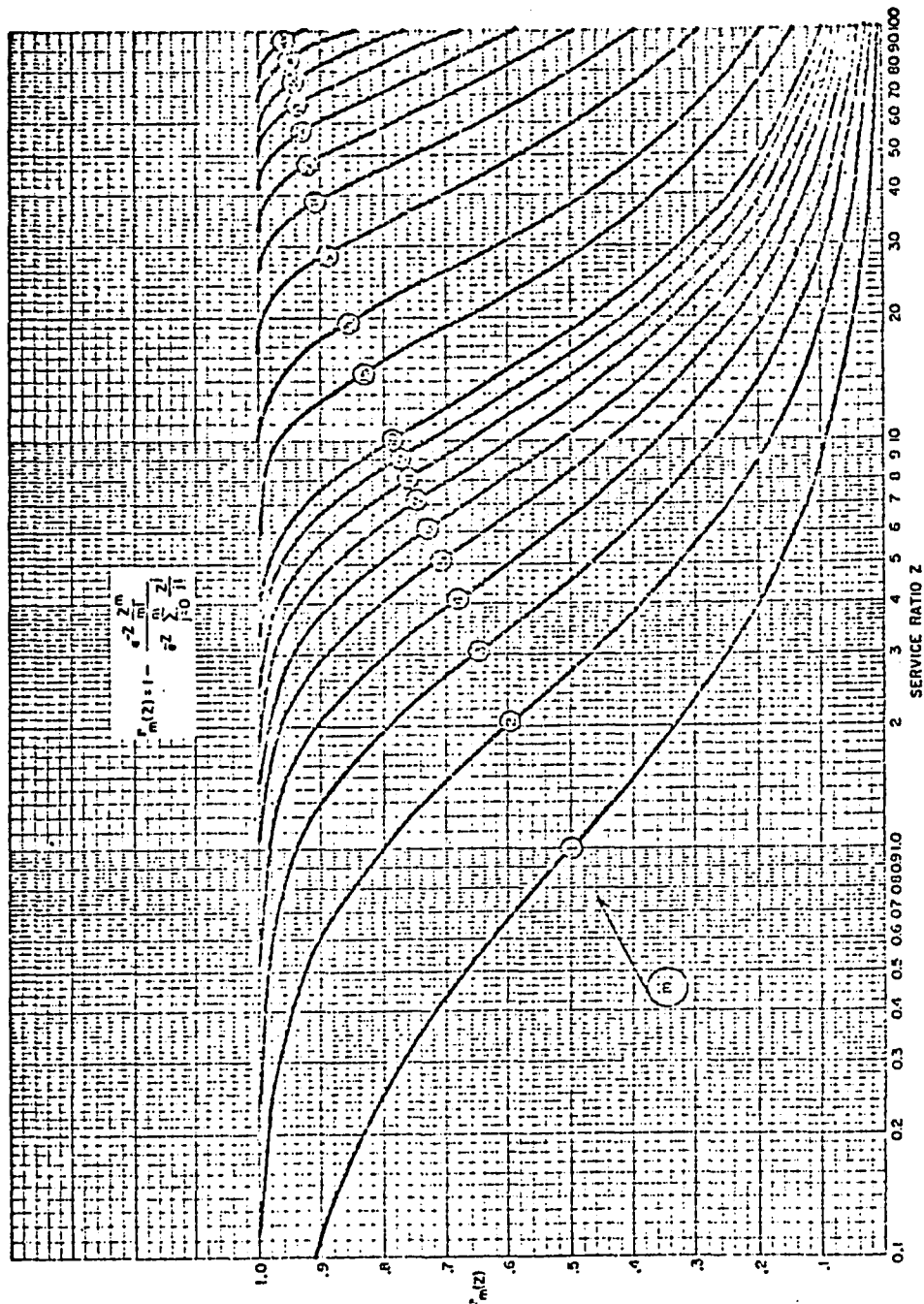


FIGURE: 4.1.5 Poisson ratio function $R_1(z)$ versus service ratio $z = T_a/T_B$ in the classical machine interference problem.

Sources: Analysis of Some Queuing Models, (New York, IBM, Ref. P20-0007-1, 1963), p. 40.

The response time for a single server model with random arrivals and an arbitrary service distribution has been obtained by Pollaczek and simplified by Khintchine. The general formula known as Pollaczek-Khintchine uses only the first two moments of the service distribution and is as it follows:

$$E[\text{time in system}] = \frac{1}{\lambda} \left[\rho + \frac{\lambda^2 V + \rho^2}{2(1-\rho)} \right] \quad 4.1.4$$

The formula can be transformed by algebraic modifications to:

$$T_q = \frac{T_p}{1-\rho} \left[1 - \rho/2 \left(1 - \frac{\sigma_p^2}{T_p^2} \right) \right] \quad 4.1.5$$

where $\frac{1}{\mu} = \bar{T}_b$ and $V = \sigma^2$.

The total response-time has been obtained in terms of T_p the service-time when the overall system is assumed to be a service-station. The only missing information necessary to obtain the expected value of T_p is the expected waiting time to the CPU itself.

Applying the machine interference results for the middle subsystem where the machines or terminals are queuing the CPU, one can obtain the remaining values. The server's (CPU) utilization $R(m)Z$ can be found from the graph provided in figure 4.1.5. However, the service ratio z is not known, since it depends on the external arrival rate. This ratio can be determined in the following manner: since all customers arriving at the various machines must eventually go through the service queue, the utilization of the serviceman can be calculated independently from z to be

$$r_m(z) = \lambda \cdot \bar{T}_s \quad 4.1.6$$

³ T. L. Saaty, Elements of Queuing Theory (New York: McGraw Hill, 1961) p. 40.

$Z \bar{T}_s = \bar{T}_a$ and $r_m(z) = \lambda \bar{T}_s$ can be substituted in the general machine interference formula

$$E(\text{time between breakdowns}) = \frac{m \bar{T}_s}{r_m(z)} - \bar{T}_a$$

The following results are thus obtained.

$$\bar{T}_w + \bar{T}_s = M/\lambda - Z \bar{T}_s \quad \text{if } \frac{\bar{T}_w}{\bar{T}_s} > 1$$

4.1.7

However, if $\frac{\bar{T}_w}{\bar{T}_s} < 1$, the simple queuing time formula may be used as a good approximation.

$$\bar{T}_w + \bar{T}_s = \frac{\bar{T}_s}{1 - \frac{(m-1)}{m} \lambda \bar{T}_s} \quad \text{if } \frac{\bar{T}_w}{\bar{T}_s} < 1$$

4.1.8

where Z is the service ratio in the machine interference model.

The behavior of the model can be shown better in terms of an example. In that example there are 20 terminals connected to the CPU. The key-in time is uniformly distributed between 5 and 15 and the print out of a message is also uniformly distributed between 2 and 7. The computer processing time is assumed to be an Erlang-2 distribution with a mean of 2 seconds. Although the CPU processing time is not exponentially distributed, the machine interference formulae are used as an approximation.

In order to obtain numerical values for the expected response-time and for the other components one may proceed as it follows:

CPU utilization is determined from equation 4.1.6

$$r_{20}(z) = \lambda \frac{\text{Inquiries}}{\text{sec}} \times 2 \frac{\text{sec}}{\text{inquiry}}$$

From this equation the service ratio z is determined, which can then be substituted in equation 4.1.7 and 4.1.8.

$$\bar{T}_w + \bar{T}_s = \frac{2}{1 - 0.95 \times 2 \times \lambda} \quad \text{if } \frac{\bar{T}_w}{\bar{T}_s} < 1$$

or

$$\frac{20}{\lambda} - 2z, \quad \text{if } \frac{\bar{T}_w}{\bar{T}_s} > 1$$

The overall inquiry service time found from equation 4.1.1 is then given as

$$T_p = 10 + T_w + T_s + 5$$

The inquiry service variance from equation 4.1.2 is likewise given as

$$\sigma_p^2 = \frac{(15-5)^2}{2} + T_w^2 + \frac{2^2}{2} + \frac{(7-3)^2}{12} = 11.7 + T_w^2$$

where σ_w^2 is approximated by T_w^2 .

Terminal utilization is given from equation 4.1.3

$$\rho = \frac{\lambda}{20} T_p$$

Finally, inquiry response time is determined from the queuing time formula of equation 4.1.4. The whole set of calculations is summarized in Table 4.1.1.

With 20 terminals (as shown on table 4.1.1) the system can accept 0.4 inquiries per second or 1.2 inquiries per minute per terminal. The response time is 30 seconds with CPU utilization of 80%. Beyond this point the response time increases at an increasing rate as shown in figure 4.1.6.

The reader will notice that the assumptions have been made above concerning the arrival distribution at the second stage of the system. (see figure 4.1.1) Burke has shown that the output of one queue with Poisson input is also Poisson.⁴ However, the general procedures to obtain the distribution of the output of queues in parallel and series is difficult to obtain analytically. In this analytical model assumptions have to be made at each stage. It is assumed that the input to the second stage is also Poisson as the input to the system.

⁴

P. J. Burke, "The Output of a Queuing System". Operations Research; Vol, 4, December 1956, pp. 699-704

Example of the results obtained
from the analytical solution

R_{20}	λ	Z	$\bar{T}_w$	$\bar{T}_p$	σ_p^2	$\sigma_p^2 / \bar{T}_p^2$	ρ	$\bar{T}_q$
0	0		0	17	11.7	0.04	0	17
0.1	0.05	200	0.21	17.2	11.7	0.04	0.04	17.5
0.2	0.10	98	0.47	17.5	11.9	0.04	0.09	18.5
0.3	0.15	65	0.80	17.8	12.3	0.04	0.13	19.2
0.4	0.20	48.4	1.20	18.2	13.1	0.04	0.18	20.2
0.5	0.25	38.2	1.60	18.6	14.3	0.04	0.23	21.5
0.6	0.30	31.2	2.65	19.7	18.7	0.05	0.30	24.2
0.7	0.35	25.9	3.20	20.3	22.0	0.05	0.35	26.0
0.8	0.40	21.6	4.80	21.8	34.7	0.07	0.44	31.2
0.9	0.45	17.6	7.30	24.3	65.	0.11	0.55	41.0
0.95	0.475	15.2	9.7	26.7	106.	0.15	0.64	54.0
0.98	0.49	13.2	12.5	29.5	168.	0.19	0.72	75.0
1.0	0.50	0	38	55	1450	0.48	1.4	-

TABLE 4.1.1

Source: Analysis of Queuing Models, p. 45.

R_{20} = CPU utilization

λ = Arrival rate

Z = Service ratio

T_w = Waiting for CPU

T_p = Total time in system excluding waiting for terminal

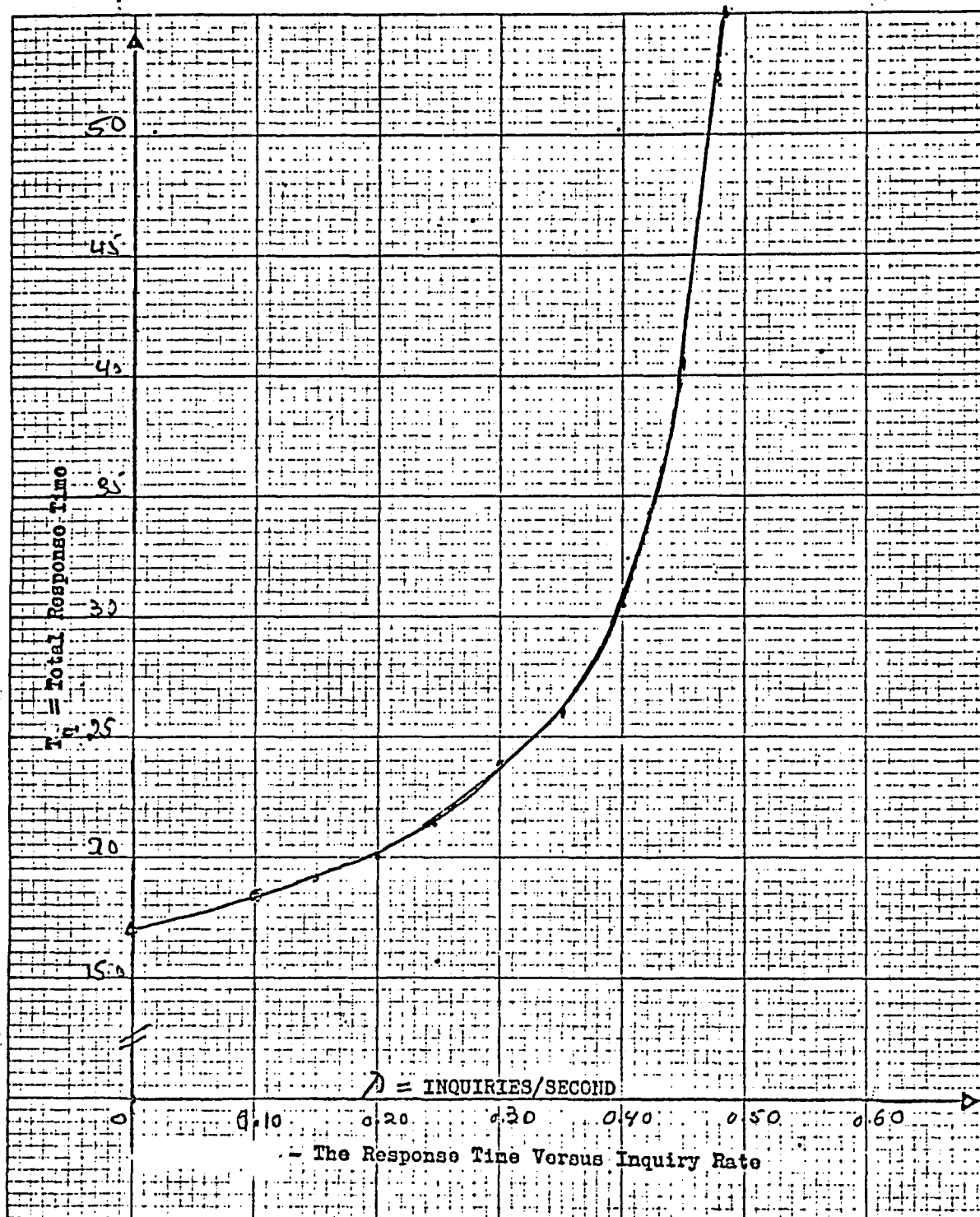
σ_p^2 = Variance

ρ = Terminal utilization

T_q = Total time in system

Expected response time versus inquiry rate

Figure 4.1.6



Source: Drawn by the author according to table 4.1.1

In solving the model, assumptions were made and approximations were used. The model was analytically solved by studying each of the subsystems and by adding up the results. Intermediate assumptions were made concerning each subsystem. The middle stage was approximated by the machine interference model where the "down" times are exponentially distributed. Even in that middle stage, the usual queuing theory formulae for arrivals from an infinite population were used where the ratio of the waiting time over the service-time at the CPU was small.

As any abstraction, the model represents only some aspects of the real world. Therefore, it is important to know the effect of assuming or neglecting certain parameters. The model is based on the "Poisson" arrival assumption and no solution is given for a non Poisson arrival. One may ask what would happen if the arrival distribution is not really Poisson. For this reason the two following hypotheses will be tested.

HYPOTHESES

- A. The expected total response time is properly obtained from the analytical solution.
- B. The arrival distribution has no effect on the expected total response time of the simulation.

SIMULATION RESULTS

Simulation runs using GPSS as described in chapter three were made to test the above hypotheses. The simulation runs are made for the same values as in the example given above in table 4.1.1. The program RL10 listed in appendix three generates Poisson arrivals randomly distributed to any of the 20 terminals. A queue is formed in front of each terminal. The units go through the stages of the model illustrated in figure 4.1.2. In the simulation, there is no need to make

assumptions at each stage since the program will take as input in stage 2 the output of stage 1. There is measurement only at the end, for the customer is interested in knowing the response time of the overall system rather than the waiting at a stage.

The First Hypothesis. The system was studied at eight different rates of arrival corresponding to the rates provided in table 4.1.1. In order to arrive at a confidence interval at each point of interest one must have a sample of independent observations. However, the simulation results being autocorrelated it is not possible to obtain the variance directly.

As it has been discussed in chapter three, the variance of autocorrelated data is not related to the population variance by the simple expression σ^2/n as with independent data. In order to avoid the problem of autocorrelation 12 independent runs were made at each of the 8 arrival rates being studied.

In each run a transient period of 50 arriving units was discarded and 100 steady state units were recorded. By making exploratory runs it was found that the steady-state was reached much before 50 arriving units. For example, the results of 100 observations recorded after 30 transient units produced the same results as 100 observations recorded after 1000 transient units. The means of the response time for each are approximately normally distributed and a confidence interval can be calculated.

It is not possible to show here the results of so many runs. However, the expected response time for each run was recorded and entered in the first column of table 4.2.3.

Although one is interested in the expected total response time, many other values are given by the program. A sample of an actual output of the program RL10 is shown in table 4.2.1 and 4.2.2. The output is divided into three sections. First, the "utilization" section gives for each of the facilities (i.e. 20 terminals called "ter 1" to "ter 20" and one CPU) such statistics as the average utilization, the average time per transaction, and the number of entries. Second, the "waiting" section gives for each of the 20 queues to the above facilities such statistics as the maximum contents, average entries, total entries, the number of non-waiting entries, the average waiting time, and the average waiting time for the units who wait. Third, the section "transit time" gives not only the mean and the variance of the response time but also the distribution of the response time. For each of the 288 runs, the author has recorded in table 4.2.3 the mean response time which can be found as the "mean argument" of the section "transit time" in table 4.2.2.

The output of the simulation runs are compared to the analytical results in table 4.2.4. Even if one is interested here in the expected total response time, table 4.2.4 also shows the utilization values, the waiting times, and the length of the queue. However, the response time has not been obtained by adding up the different pieces but by measuring the difference of time between the arrival and the departure of a unit. The other values are presented to help identifying the area of great differences as it will be discussed later. Both response times are graphed in figure 4.2.1. The reader will notice that there is a major difference in the results when the arrival rate approaches 1.0 unit per second or when the CPU utilization approaches 1.

Section one.

UTILIZATION

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN	SEIZING TRANS. NO.	PREFEMPTING TRANS. NO.
TER1	.667	7	17046.140	2	
TER2	.594	6	17712.664	4	
TER3	.246	3	14706.664	21	
TER4	.414	4	18535.250		
TER5	.932	8	20434.750	14	
TER6	.166	2	9563.750		
TER7	.760	7	19433.710		
TER8	.845	8	18499.125	24	
TER9	.157	1	24516.000		
TER10	.563	5	20141.597		
TER11	.454	4	20375.000		
TER12	.556	4	24587.000		
TER13	.473	5	17458.398		
TER14	.640	6	19080.500		
TER15	.472	4	21117.500	7	
TER16	.756	6	22539.164		
TER17	.908	8	20307.875	19	
TER18	.640	8	14307.000	8	
TER19	.906	7	20612.285	12	
TER20	.687	6	20491.664		
CPU	.932	100	1667.549	12	

Section two.

WAITING

QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
TER1	1	.113	7	3	42.8	2897.856	5071.250		1
TER2	2	.171	7	4	57.1	4389.140	10241.332		1
TER3	1	.000	3	3	100.0	.000	.000		
TER4	1	.033	4	2	50.0	1512.000	3024.000		
TER5	2	.558	8	2	25.0	12489.625	16651.500		
TER6	1	.072	1	.0	.0	499.000	499.000		
TER7	3	.562	6	3	50.0	16775.332	33550.664		
TER8	3	.703	8	1	12.5	15715.250	17960.285		1
TER9	1	.000	1	1	100.0	.000	.000		
TER10	1	.074	4	2	50.0	3339.250	6678.500		
TER11	1	.154	3	1	33.3	9231.332	13947.000		
TER12	1	.014	4	3	75.0	641.750	2567.000		
TER13	1	.000	4	4	100.0	.000	.000		
TER14	1	.110	5	2	39.9	3939.399	6565.664		
TER15	1	.116	4	2	50.0	5200.250	10400.500		
TER16	2	.883	5	.0	.0	31601.597	31601.597		
TER17	2	.509	7	1	14.2	13012.140	15180.832		
TER18	2	.340	8	3	37.5	7610.875	12177.398		1
TER19	2	.309	7	3	42.8	7905.000	13931.750		
TER20	3	1.086	7	1	14.2	27764.855	32392.332		1
CPU	8	2.332	101	14	13.8	4130.105	4794.722		1

AVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

TABLE: 4.2.1

SOURCE: OUTPUT OF THE PROGRAM RL10 LISTED IN APPENDIX THREE

THIS IS ONLY ONE EXAMPLE OF THE OUTPUT PRODUCED.

IN THIS EXAMPLE THE INTER-ARRIVAL TIME IS 2 SEC.

THE ARRIVAL RATE IS .50 PER SEC.

THE ARRIVAL DISTRIBUTION IS POISSON

Section Three.

THE TRANSIT TIME IN THE SYSTEM

TABLE XTIME		MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS		NON-WEIGHTED
ENTRIES IN TABLE						
100		33106.136	16800.000	3310614.000		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-1.970
5000	0	.00	.0	100.0	.151	-1.672
10000	0	.00	.0	100.0	.302	-1.375
15000	4	3.99	3.9	96.0	.453	-1.077
20000	20	19.99	23.9	76.0	.604	-.780
25000	20	19.99	43.9	56.0	.755	-.482
30000	11	10.99	54.9	45.0	.906	-.184
35000	11	10.99	65.9	34.0	1.057	.112
40000	10	9.99	75.9	24.0	1.208	.410
45000	3	2.99	78.9	21.0	1.359	.707
50000	4	3.99	82.9	17.0	1.510	1.005
55000	4	3.99	86.9	13.0	1.661	1.303
60000	2	1.99	88.9	11.0	1.812	1.600
65000	4	3.99	92.9	7.0	1.963	1.898
70000	2	1.99	94.9	5.0	2.114	2.196
75000	3	2.99	97.9	2.0	2.265	2.493
80000	1	.99	98.9	1.0	2.416	2.791
85000	0	.00	98.9	1.0	2.567	3.088
90000	0	.00	98.9	1.0	2.718	3.386
OVERFLOW	1	.99	100.0	.0		
AVERAGE VALUE OF OVERFLOW		92696.00				

44

NOTE: IT IS NOT POSSIBLE TO SHOW IN THIS DISSERTATION ALL THE 596 PAGES OF MEANINGFUL OUTPUT OF THE PROGRAM RL10.
TABLE 4.2.1 AND 4.2.2 ARE SHOWN HERE TO GIVE THE READER SOME INFORMATION ABOUT THE TYPE OF OUTPUT PRODUCED.

TABLE: 4.2.2

SOURCE: OUTPUT OF THE PROGRAM RL10 LISTED IN APPENDIX THREE

THIS IS ONLY ONE EXAMPLE OF THE OUTPUT PRODUCED.

IN THIS EXAMPLE THE INTER-ARRIVAL TIME IS 2 SEC.

THE ARRIVAL RATE IS .50 PER SEC.

THE ARRIVAL DISTRIBUTION IS POISSON

**THE EXPECTED TOTAL RESPONSE TIME
FOR 288 RUNS**

MEAN ARRIVAL RATE SEC.	ARRIVAL DISTRIBUTION											
	POISSON				UNIFORM				NORMAL			
0.10	18.7	17.3	18.4	17.6	17.9	18.8	18.2	18.1	17.5	17.2	17.7	17.9
	17.5	18.0	18.5	18.3	18.7	17.6	17.3	17.9	16.6	17.4	17.3	17.3
	18.3	17.6	19.2	17.8	18.3	17.9	17.0	18.3	18.2	18.4	17.9	17.7
0.20	20.9	21.6	19.4	18.6	18.8	19.7	19.4	18.8	19.1	17.9	18.6	17.3
	18.7	19.8	19.8	22.4	20.6	20.8	18.8	20.3	18.9	18.9	18.4	19.0
	20.9	20.2	18.7	19.6	18.0	20.1	19.0	21.3	19.5	18.6	18.9	19.7
0.30	22.9	23.3	25.8	21.8	22.4	21.6	19.8	20.5	22.4	22.9	20.4	20.3
	20.4	25.1	22.5	27.1	18.9	27.3	22.6	22.4	20.8	20.5	21.3	22.3
	23.1	20.2	21.0	22.4	23.3	20.8	19.9	21.6	21.2	23.6	21.8	22.8
0.40	44.1	32.4	27.2	26.4	33.5	24.5	20.9	26.2	27.1	24.3	22.1	23.1
	28.2	21.0	23.1	23.1	26.5	36.2	24.5	27.2	23.9	27.2	26.0	24.6
	28.1	29.6	26.1	33.3	25.3	26.4	32.7	37.9	21.5	27.3	26.7	27.6
0.45	41.6	27.7	32.1	43.5	32.1	29.7	26.6	32.5	32.7	26.6	26.9	29.0
	31.2	27.1	37.8	38.4	29.1	27.3	26.4	25.0	30.8	31.8	23.7	25.6
	31.3	24.4	27.7	29.3	50.0	32.4	29.8	29.6	26.1	30.2	29.6	34.2
0.50	31.9	33.1	57.3	28.9	38.3	29.3	46.4	37.4	40.7	32.5	32.2	29.0
	39.1	37.8	30.9	41.8	43.8	39.3	33.7	44.8	32.3	46.9	41.0	28.2
	59.6	39.7	59.1	70.2	58.6	40.6	32.4	33.7	31.3	39.6	34.3	39.4
0.555	47.5	39.8	35.2	61.7	55.3	33.1	41.1	43.2	34.4	43.7	34.7	60.0
	40.0	29.4	36.2	25.4	61.5	46.8	40.2	54.0	39.1	60.6	55.2	41.0
	90.6	54.7	37.4	56.5	51.2	48.3	45.6	52.2	39.5	45.3	59.7	62.1
0.666	65.6	106.0	43.7	49.2	57.3	65.0	83.4	73.9	69.3	45.5	76.6	83.8
	95.8	91.1	60.2	57.2	64.2	72.5	56.2	80.8	76.2	70.0	67.8	69.4
	71.2	74.9	63.9	71.5	77.8	61.4	45.6	56.7	64.1	78.2	51.0	71.3

NOTE: EACH VALUE SHOWN IN THIS TABLE IS THE AVERAGE OF A RUN MADE OF 100 NON TRANSIENT OBSERVATIONS.
 IN EACH OF THE 288 RUNS THE SYSTEM WAS FIRST LOADED WITH 50 ARRIVALS.
 12 DIFFERENT SEQUENCES OF RANDOM NUMBERS WERE USED FOR THE ARRIVAL;
 EACH SEQUENCE WAS OBTAINED BY SELECTING A SEED AT RANDOM FOR EACH
 RATE AND FOR ALL THREE TYPES OF ARRIVAL DISTRIBUTIONS.
 THE SAME SEQUENCES HAVE BEEN USED FOR ALL 8 LEVELS OF THE ARRIVAL.
 WITHIN A CELL THE RESULTS ARE INDEPENDENT.

TABLE: 4.2.3

SOURCE: SUMMARY OF THE TOTAL RESPONSE TIME OBTAINED BY 288 RUNS OF THE
 PROGRAM RL10 LISTED IN APPENDIX THREE.

ANALYTICAL RESULTS VS SIMULATION RESULTS

INTER ARRIVAL TIME (SEC)		ARRIVAL RATE		UTILIZATION OF CPU		WAITING FOR CPU		TERMINAL UTILIZ.		RESPONSE TIME		RESPONSE TIME		LENGTH OF AVERAGE CPU		QUEUE CPU MAXIMUM
ANAL.	SIM.	ANAL.	SIM.	ANAL.	SIM.	ANAL.	SIMU.	ANAL.	SIM.	EXCL. ANAL.	TERM. SIM.	INCL. ANAL.	TERM. SIM.	ANAL.	SIM.	SIM.
10	10	0.10	0.10	0.2	0.197	0.47	0.324	0.09	0.083	17.5	17.25	18.5	18.109	0.047	0.035	2.0
5	5	0.2	0.2	0.4	0.395	1.20	0.928	0.18	0.161	18.2	18.48	20.2	20.012	0.240	0.194	6.0
3.33	3.33	0.3	0.3	0.6	0.606	2.65	1.749	0.30	0.321	19.7	18.50	24.2	22.915	0.795	0.518	7.0
2.5	2.5	0.4	0.4	0.8	0.783	4.80	4.017	0.44	0.475	21.8	22.67	31.2	28.557	1.90	1.771	10.0
2.22	2.22	0.45	0.45	0.9	0.895	7.30	5.563	0.55	0.428	24.3	24.00	41.0	32.724	19.0	2.578	10.0
2.0	2.0	0.5	0.5	1.0	0.961	38.0	8.493	1.0	0.708	55.0	LARGE	LARGE	43.793	LARGE	4.281	13.0
1.8	1.8	0.555	0.555	1.0	0.967	LARGE	8.512	1.0	0.916	LARGE	LARGE	LARGE	46.206	LARGE	4.408	14.0
1.5	1.5	0.666	0.666	1.0	0.999	LARGE	15.601	1.0	0.965	LARGE	LARGE	LARGE	71.010	LARGE	8.516	15.0

ANAL. : RESULTS OF ANALYTICAL SOLUTION

SIM: SIMULATION RESULTS

LARGE: MEANS THAT THE VALUE APPROACHES INFINITY

TABLE: 4.2.4

SOURCE: PREPARED BY THE AUTHOR FROM THE RESULTS OF TABLE 4.1.1
AND FROM THE SIMULATION RUNS

The analytical results imply that the mean response time curve shown on figure 4.2.1 approaches asymptotically a vertical line at an arrival rate below 0.5 inquiry per second. The simulation results show that the expected response time curve approaches asymptotically a vertical line at 0.75 inquiry per second.

It should be noted here that as shown in appendix three the results of the simulation model did not start with an empty system which will have produced results further apart from the analytical results. The model was run until a steady state had been reached, only at that point were the statistics accumulated.

As one recalls, the analytical solution was obtained by adding the waiting time and the service time at each stage. One can see by looking at table 4.2.4 that the element that seems to differ the most between the analytical results and the simulation results is the waiting time at the CPU itself. This corresponds to the waiting time at the stage which was approximated by the machine interference model. It is interesting to note that in the area where the ratio of the average waiting time to the CPU over the average service time was less than one (i.e. $\frac{\bar{W}_c}{\bar{S}_c} < 1$) the results of the analytical solution and of the simulation are statistically the same. In this region the machine interference model was not used.

Confidence intervals for the total response time were calculated using Student's Statistics for the 8 arrival rates of interest at 99% level and 95% level. The results are summarized in table 4.2.5 and plotted in figure 4.2.1. A sample of the actual output of the program used to do the calculation is given in table 4.2.6. One can easily see that for an arrival rate of over 0.45 inquiry per second, the analytical results are way out of the 99% confidence interval.

CONFIDENCE LIMITS FOR RESPONSE TIME

ARRIVAL		UTILIZATION OF CPU		TOTAL TIME		RESPONSE STD. ERR. OF MEAN	CONFIDENCE INTERVAL FOR POP. MEAN	
TIME	RATE	ANAL.	SIMUL. * MEAN OF MEANS	ANAL.	SIMUL * MEAN OF MEANS		95% LEVEL	99% LEVEL
10	110	.2	0.197	18.5	18.109	0.162857	17.7429 - 18.6042	17.5958 - 18.604
5	.2	.4	0.395	20.2	20.012	0.326897	19.4145 - 20.8522	19.1213 - 21.145
3.33	.3	.6	0.606	24.2	22.915	0.614245	21.6160 - 24.3174	21.0651 - 24.868
2.5	.4	.8	0.783	31.2	28.557	1.76127	24.6770 - 32.4230	23.0974 - 34.002
2.22	.45	.9	0.895	41.0	32.724	1.79238	28.7170 - 36.5997 ***	27.1094 - 38.207
2.0	.50	1.0	0.961	LARGE **	43.793	2.52822	28.8989 - 40.0178 ***	26.6314 - 42.285
1.8	.555	1.0	0.967	LARGE **	46.206	5.13776	34.9023 - 57.4977 ***	30.2943 - 62.105
1.5	.660	1.0	0.995	LARGE **	71.010	5.41217	58.9572 - 82.7595 ***	54.1031 - 87.613

* THE VALUE SHOWN IS THE AVERAGE VALUE OF 12 RUNS.
EACH RUN HAS 100 STEADY STATE OBSERVATIONS.

** APPROACHES INFINITY AS UTILIZATION APPROACHES 1.0.
OR AS ARRIVAL RATE APPROACHES 1.0

*** THE RESULTS OF MATHEMATICAL FORMULAE GIVES A RESPONSE
TIME OUTSIDE THE CONFIDENCE INTERVAL.

NOTE : ALL THE VALUES SHOWN ON THIS TABLE ARE THE MEAN
OF ONE SIMULATION RUN OF 100 OBSERVATIONS EXCLUDING
TRANSIENT OBSERVATIONS.
100 OBSERVATIONS FOR 1 ENTRY IN THE TABLE
12 ENTRIES PER CELL
9 LEVELS I.E. 8 ARRIVAL AVERAGES
3 TREATMENTS OR 3 ARRIVAL PATTERN
24 CELLS
288 ENTRIES
28,800 OBSERVATIONS

TABLE: 4.2.5

SOURCE: PREPARED BY THE AUTHOR FROM THE RESULTS SHOWN ON TABLE 4.2.1 AND 4.2.2

THE SIMULATION RESULTS ARE FROM THE PROGRAM RL10 LISTED IN APPENDIX THREE

THE CONFIDENCE LIMITS HAVE BEEN PRODUCED BY THE PROGRAM CONLM1 LISTED IN APPENDIX FOUR

TOTAL RESPONSE TIME VERSUS INQUIRY RATE

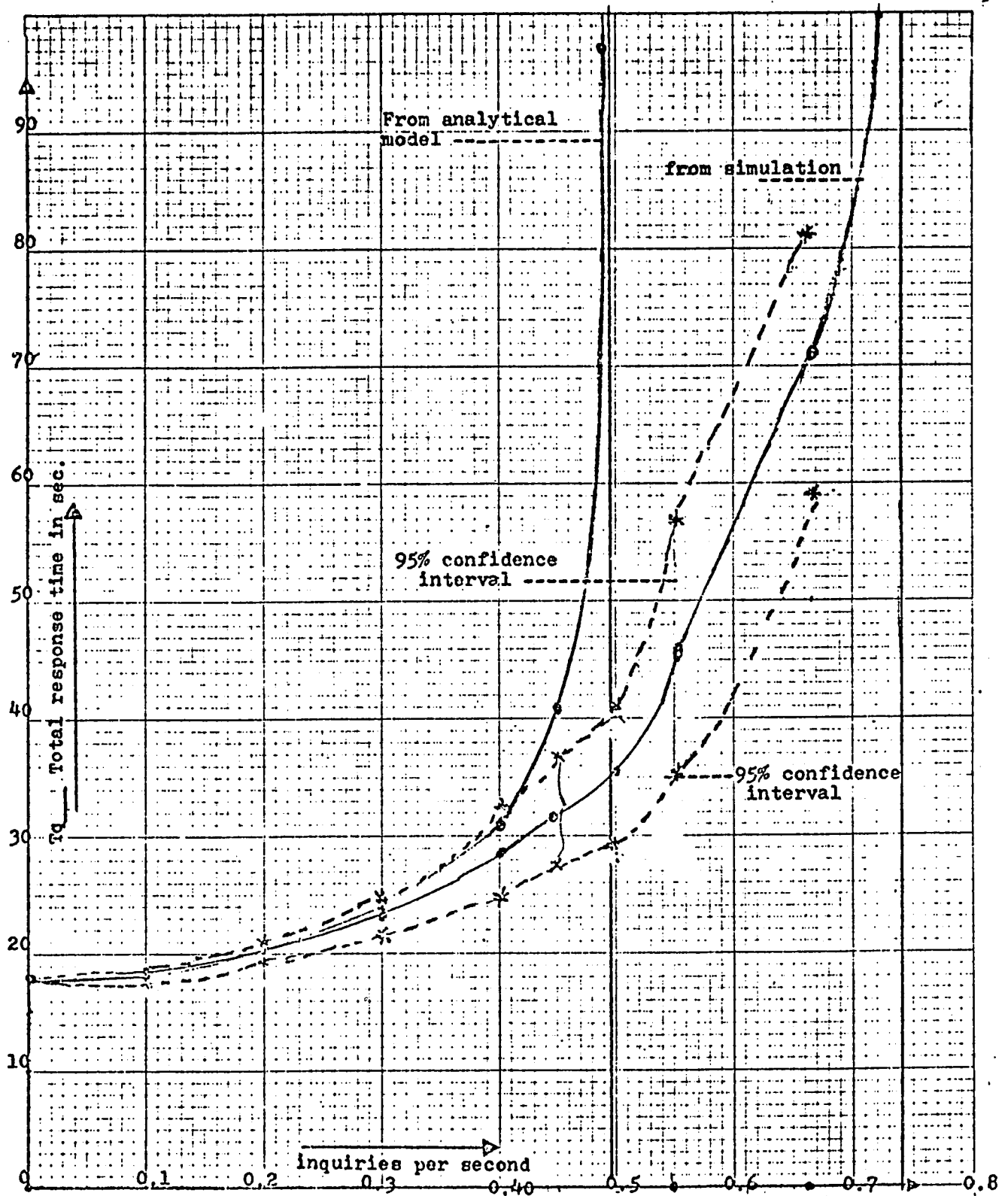


Figure 4.2.1

SAMPLE OUTPUT OF THE PROGRAM USED TO CALCULATE THE CONFIDENCE LIMITS.

9901 DATA 47.5,39.8,35.2,61.7
 9902 DATA 40.0,29.4,36.2,25.4
 9903 DATA 90.6,54.7,37.4,56.5
 RUN
 CONLM1 1610

VALUES OF SAMPLE STATISTICS:

SIZE OF SAMPLE	12	----- 12 runs of 100 observat
SAMPLE MEAN VALUE	46.2	
VARIANCE OF SAMPLE	290.363	
SAMPLE STD DEVIATION	17.04	
ESTIMATED POPN STD DEV	17.7977	
STANDARD ERROR OF MEAN	5.13776	

CONFIDENCE LIMITS ON POPULATION MEAN:

CONF LEVEL	LOWER LIM	UPPER LIM
50	42.6167	49.7833
75	39.961	52.439
90	36.9771	55.4229
95	34.9023	57.4977
99	30.2943	62.1058
99.9	23.6397	68.7604
99.99	16.4925	75.9075
99.999	8.58067	83.8194

DONE AT 1611

TABLE 4.2.6

SOURCE: output of the program CONLM1 listed in appendix four.

It is easy to conclude that the results fail to confirm the first hypothesis. The response time does not increase as fast as suggested by the analytical techniques. The reason for such a difference may be that the input to the second stage of the model i.e. the CPU is not exponentially distributed as it has been assumed to fit the machine interference model. The later model assumed exponentially distributed "down" times on each of the 20 terminals. However, since a queue is formed in front of each terminal followed by a uniform service time, the equivalent of the "down" times are not necessary exponentially distributed. The reader will notice that in the simulation runs there is no need to make assumptions at the second and each subsequent stage as in the analytical solution.

Testing the Second Hypothesis. An important factor often mentioned in textbooks and publications is that a special study should be undertaken to make sure that the arrival rate is Poisson if one wants to use it. In this dissertation the author uses another approach i.e. sensitivity analysis of the assumptions. One would ask What if the assumptions are not true? Should we investigate the real arrival pattern?

It is obvious that there is no need to investigate the exact value of a factor if this factor has no effect on the system. For this reason the author's approach is to determine in advance the effect of the mean of the arrival distribution on the system.

In order to do so, the author repeated the experiments described above for two other distinctive arrival patterns i.e. normal and uniform arrival. The means of the three distributions are the same. The uniform distribution varies from 0 to $2\bar{X}$ where $\bar{X}$ is the mean. The standard deviation used for the normal distribution is in this case $\bar{X}/5$. In each case, as before, 12 runs of 100 steady state arrivals were done for each of the

8 levels of interest. The results obtained are summarized in table 4.2.3. There are 288 runs of 100 observations i.e. 28,800 observations in all not including the transient period. There is also enough information to test the interaction for each cell in the table has 12 observations.

An analysis of variance was done to determine the effect of the treatment (arrival distribution) on the response time. The arrival distributions and the arrival means are both fixed factors. There are 3 levels for the first and 8 levels for the second factor and 12 independent observations in each cell as shown in table 4.2.3.

The arrival means have an obvious effect on the mean response time. However, here one is interested in testing the effect of the arrival distribution and the interaction effect. The results of the analysis of variance are shown in table 4.2.7. The small F value shows well that the arrival distribution has no significant effect on the total response time (even at a low 73% confidence level) and that the interaction i.e. the combined effect of arrival rate and arrival pattern is almost non-existent. We do not reject the second hypothesis.

This conclusion implies that there is no need to search for the true arrival pattern since it has little effect. Even if the arrival to the system exhibits a departure from the Poisson assumption the expected response time will not be affected.

CONCLUSION

In this chapter the author has reviewed and tested by simulation a classic real-time model. He has shown not only that the system does not behave as prescribed by the analytical solution but also the original conclusion that the arrival distribution has little effect on the expected total response time.

EFFECT OF THE ARRIVAL PATTERN ON THE RESPONSE TIME IN A REAL TIME MODEL

ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	74925.938	7.	10703.703	191.956
S(A)	4907.000	88.	55.761	
B	268.438	2.	134.219	1.980
AB	486.063	14.	34.719	0.512
BS(A)	11931.188	176.	67.791	

GRAND MEAN = 34.192

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	74915.250	7.	10702.176	191.928
S(A)	4907.000	88.	55.761	
B	257.766	2.	128.883	1.901
AB	495.891	14.	35.421	0.523
BS(A)	11931.188	176.	67.791	

GRAND MEAN = 34.192

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3
1	18.100	18.000	17.592
2	20.050	19.633	18.733
3	22.967	21.758	21.692
4	28.550	28.483	25.117
5	32.675	30.875	28.933
6	43.775	40.025	35.617
7	46.200	47.708	47.942
8	70.558	66.233	69.100

NOTE: A = ARRIVAL MEANS ; B = ARRIVAL PATTERN

A AND B ARE FIXED FACTORS ; A IS MAIN EFFECT AND IS CONFOUNDED WITH PLOT

TABLE: 4.2.7

SOURCE: OUTPUT OF THE PROGRAM FINAGAV LISTED IN APPENDIX FIVE

Simulation has been used successfully to arrive at this conclusion because there is no need for assumptions to be made at each stage as in the analytical solution. Furthermore, a simulation model, once running, can easily be changed to analyze the assumptions at the first stage. It is obviously less expensive to run sensitivity analysis on the assumptions than to field test them.

Sensitivity analysis could be performed on any of the other parameters of the model. However, simulation experiments being very expensive, the author elected to allocate limited resources to the analysis of the classical queuing theory assumption, i.e. the Poisson arrival distribution.

QUEUEING MODEL OF DIRECT ACCESS METHOD

The purpose of this chapter is to test the validity of the solution of a queueing model for direct access storage devices and to test by simulation the sensitivity of the model to a departure from the classical Poisson arrival assumption.

The retrieval of information from mass storage devices is a major function of a real-time system. In the previous chapter, it has been assumed that this operation was included in the service function of the central processing unit. However, since the accessing of a mass storage device is a major factor affecting a real-time system, it will be studied individually in this chapter.

The first part of the chapter will develop a model and apply the classic queueing theory formulae to solve the model with restricted arrival assumptions. A model similar to the one presented in this chapter has already been presented¹. However, it applies to a different type of device. The measure of efficiency and the solution are also different as it will be discussed later. In the second part, simulation experiments will be made on the model with the similar assumptions. The effect of a departure from the "Poisson" arrival assumption on the expected response time will be tested. Independent experiments will be repeated for different arrival distributions such as Poisson, Normal, and Uniform.

A DIRECT ACCESS SYSTEM

The main characteristic of a direct or random access storage device is that it makes it possible to store

¹ Abate, Joseph, Dubner, Harvey, and Weinberg B. Sehdon, "Queueing Analysis of the IBM 2314 Disk Storage Facility," Journal of ACM, Vol. 15, No. 4. (October 1968), pp. 578-587.

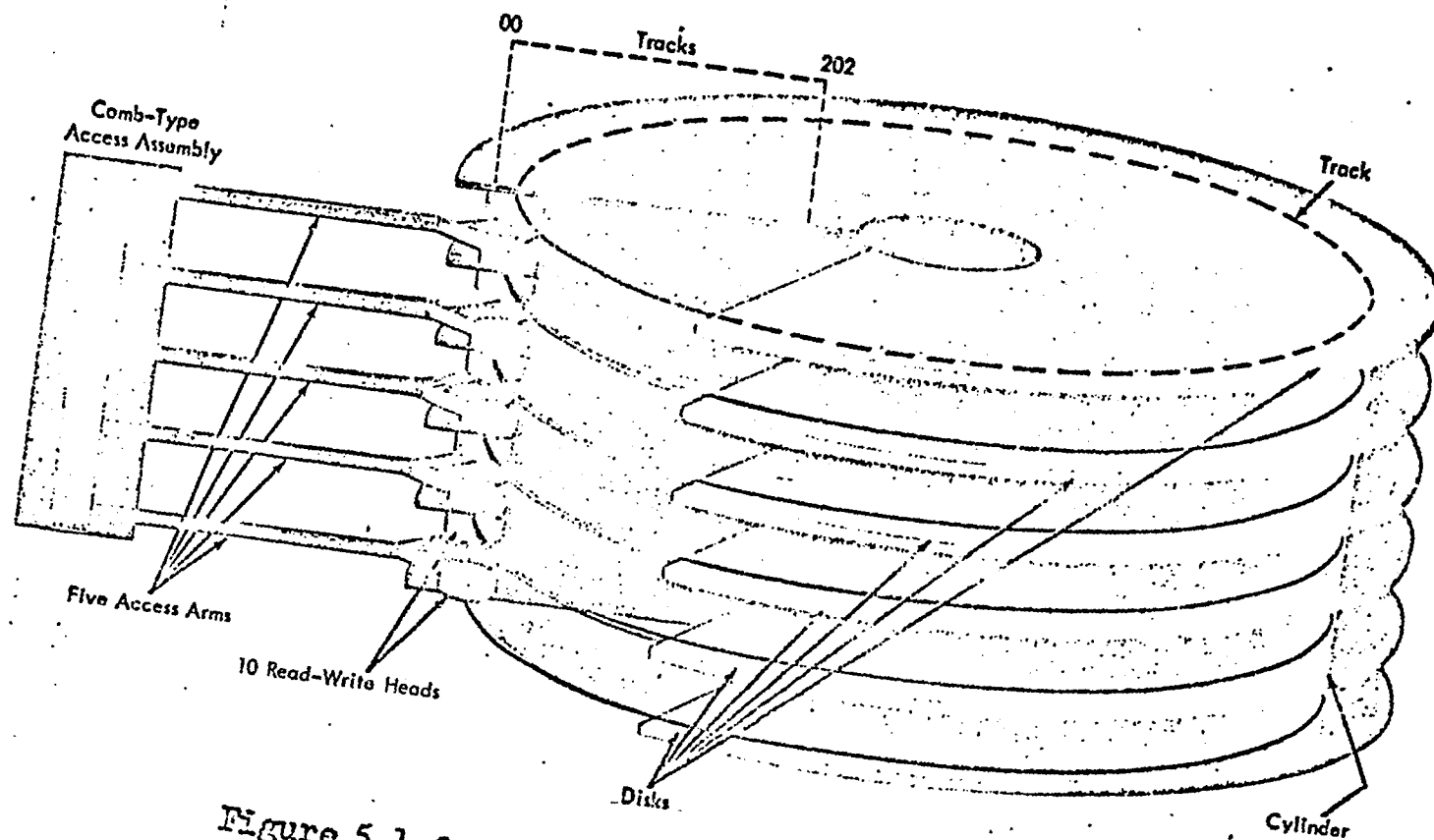


Figure 5.1.0. IBM 2311 Access Mechanism
 Source: IBM 360 Direct Access Storage Devices,
 (New York: IBM, ref: C 20-1649-2, 1969)
 P. 9.

information in such a way that it can later be retrieved randomly. By random retrieval it is meant that the position from which the information is to be obtained is independent of the position from which the previous information was obtained. There are many types of random access storage devices. However, for concreteness, this analysis applies to the IBM 2311 disk units.

Each IBM 2311 disk drive contains one removable IBM 1316 disk pack². Each pack is divided into 200 cylinders and each cylinder contains 10 tracks. There are 10 read-write arms working in parallel so that at one position there is a head for each surface as shown on figure 5.1.0.

The positioning of the head at any cylinder provides access to the 10 tracks of that cylinder. A variable number of records can be stored on each track depending on the length of the records. Each disk unit is attached to the CPU via a control unit and a physical channel. Up to eight disk units can be attached to the same control unit.

In order to retrieve information from the disk unit one must consider these operations: seek time, rotation time, and transmission time. The seek time is the time required to position the access mechanism of the disk at the desired cylinder. The arms remain in position until the unique channel is free for transmission. Once the channel becomes free the disk which rotates constantly must complete the rotation to the proper record on the track. The latter operation is called the rotation time or the rotation delay. Once the

² IBM 360 Direct Access Devices, (New York: IBM, ref: C20-1649-2, 1968), p. 8.

record is located on the track the transmission takes place. The data can be transmitted from only one disk unit at a time since there is only one control unit and one channel. However, the arm of the other disks may position themselves concurrently (1.2. seek). The seek operation is slow compared to the transmission of data.

THE MODEL TO BE STUDIED

The model necessary to represent the above operation is a two-phase model as represented on figure 5.1.1. Phase one represents the seek operation where there are k single-servers or disks in parallel. Phase two represents the rotation and transmission operations where there is only one channel.

One is interested in two measures of efficiency : the total response time of the system and the total waiting in the system. In order to obtain the values of these random variables we must study each component of the model by following a request in the system.

The access to the system begins with a request to read or write a particular record on or from a particular disk unit. If the disk is free the arm is positioned to the proper cylinder. This seek time is represented by the random variable T_1 . If the disk is busy the request waits in the queue Q_1 for the disk to become free. The waiting time is represented by the random variable W_1 . Now the request must wait in queue Q_2 for the channel to be free. This second waiting time is represented by the random variable W_2 . When the channel becomes available for our request there is a

delay represented by the random variable R . The transmission time is a function of the length of the record and is represented by the random variable D . The service time in the second phase is obtained as the sum of the two random variables.

$$T_2 = R + D$$

where $D = F(L)$

The total response time T is then the sum of four random variables.

$$T = W_1 + T_1 + W_2 + T_2$$

The total waiting time W is the sum of two random variables

$$W = W_1 + W_2$$

The total waiting time is then the time spend in the system in addition to the service time. We define a system with "perfect service" or "near-perfect service" as a system into which the time spent in excess of service is very small and less than a predetermined value established in the decision rule.

Assumptions.

In order to analyze the model the following assumptions are made:

1. The requests arrive according to a Poisson process with mean λ .
2. The requests are randomly distributed to the K disks.
3. There is no limit on the size of any queue.
4. The service time in the seek station (T_1) is exponentially distributed with mean 74.25 milliseconds.
5. The rotation delay R is uniformly distributed from 0 to 25 milliseconds.³
6. The transmission time D is exponentially distributed with mean 0.64013 millisecond.⁴

³ The figures are obtained from appendix six.

⁴ This is possible by having the length of the records exponentially distributed with mean 100. (see appendix six)

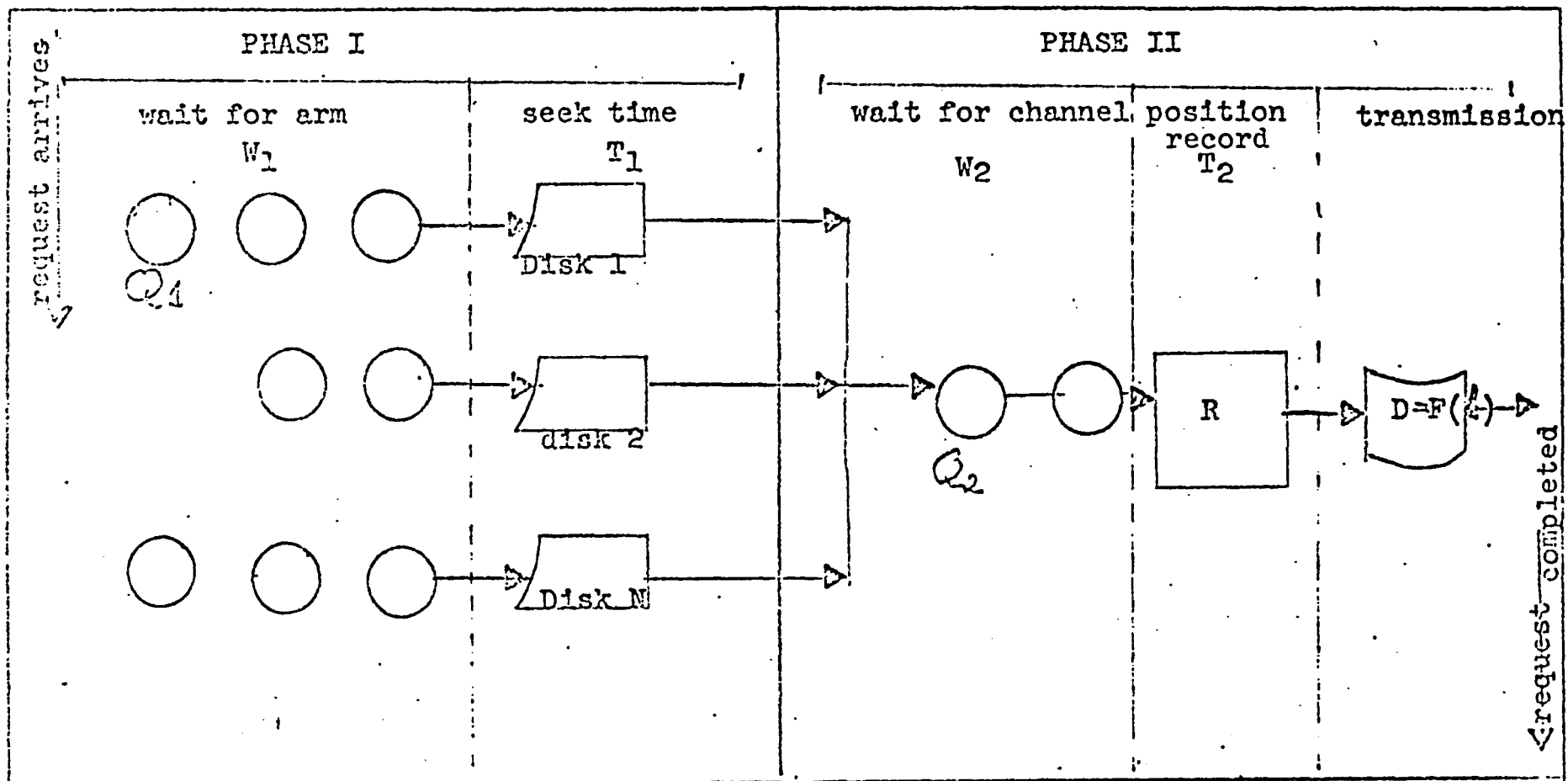


Figure 5.1.1 Queuing Model for a System with K disks and one Channel.

Source: Developed by the author

7. The random variables R and D are independent.
8. The system is studied at steady state.

Justification of The Assumptions. The implication of the Poisson arrival assumption will be analyzed later by simulation. The second assumption implies that all the K disks receive the same rate of seeks i.e. the probability of accessing any disk is $1/K$. This is possible if the records are distributed in the same fashion over all the K disks. Even if we do not use the congestion as the measure of efficiency we must analyze the relevance of the assumption of infinite queue length. The size of the waiting line is limited by the hardware. However, it can be made large enough to have no effect on the system especially when one considers a system into which the expected waiting time will be made very small as we shall discuss later.

The service time distribution of the seek operation is approximated to an exponential distribution. It is, in fact, a function of the number of tracks traveled for the access. The uniform distribution for completing the current rotation is realistic since one can be at any point on the track. The last distribution, the transmission time, can be of any shape since it is a function of the record length. In this case we assume the latter to be exponentially distributed. However, as we shall see, the analysis would be valid even if D were not exponential. The assumption of R and D being independent is very realistic.

ANALYTICAL SOLUTION

Since the model described above is in fact a sub-model of a larger real-time model, our solution is to find for each configuration the expected arrival rate which can be accepted while maintaining "perfect service".⁵ This arrival rate is then our criterion of efficiency.

The procedure is to obtain an equation for the total expected waiting time $\bar{W}$ and to find the largest λ so that the $\bar{W}$ is less than a small quantity E . The value of E is the maximum acceptable average time spent in excess of the average service time so that we can consider having a "perfect service" system. We shall set up E as 5% of the expected total service time.

The expected service time is:

$$\bar{T} = \bar{T}_1 + \bar{T}_2$$

when

$$\bar{T}_2 = \bar{R} + \bar{D}$$

$$\begin{aligned}\bar{T}_2 &= 12.5 + 0.64013 \\ &= 13.14013 \text{ milliseconds}\end{aligned}$$

$$\bar{T}_1 = 74.25$$

Therefore

$$\bar{T} = 87.39013$$

$$E = 5\% \text{ of } \bar{T} = 4.369 \text{ accesses per millisecond}$$

The total expected waiting time $\bar{W}$ can be obtained by adding the expected time for both queues.

$\bar{W} = \bar{W}_1 + \bar{W}_2$ where $\bar{W}$ is the expected value of the random variable W .

⁵ The solution presented here is different than the one given in reference 1. The latter is concerned with the response time for a given arrival rate and "fixed module of eight disks. (IBM 2314)

Since the service time in the first station i.e. seek operation is exponentially distributed we have⁶

$$\bar{W}_1 = \frac{\lambda_1}{\mu_1(\mu_1 - \lambda_1)}$$

when $\lambda_1 = \lambda/K$

K = number of disks in the system

$$1/\mu_1 = \bar{T}_1 = 74.25 \text{ milliseconds.}$$

The service time at the second station T_2 is made of two random variables R and D . Since T_2 is not exponentially distributed we can find the waiting time using the Pollaczek-Khintchine formula⁷

$$\bar{W}_2 = \frac{\lambda_2}{\mu_2^2} \cdot \left[\frac{\mu_2^2 V + 1}{2(1 - \lambda_2/\mu_2)} \right]$$

where

$$\lambda_2 = \lambda$$

$$1/\mu_2 = \bar{T}_2$$

$$V = \text{variance of } T_2 = \overline{T_2^2}$$

The only information missing is V .

Since the random variables R and D are independently distributed we can obtain the variance as follows:

$$V = \sigma_{T_2}^2 = \sigma_R^2 + \sigma_D^2$$

$$\sigma_R^2 = \frac{(b-a)^2}{12} = \frac{25^2}{12} = 52$$

when b and a are the limits of the uniform distribution.

⁶ Harvey M. Wagner, Principles of Operations Research, Englewood Cliffs, N.J.: Prentice Hall, 1969) p. 857.

⁷ IBID.

$\sigma_D^2 = 1/\mu_D = \bar{D}^2$ since D is exponentially distributed
when $1/\mu_D =$ expected service time

$$\sigma_D^2 = (0.64013)^2 = 0.40976$$

$$V = \sigma_R^2 + \sigma_D^2 = 52.40976$$

We then have

$$\bar{W} = \bar{W}_1 + \bar{W}_2$$

$$= \left[\frac{\lambda_1}{\mu_1(\mu_1 - \lambda_1)} \right] + \left[\frac{\lambda_2 \left(\mu_2^2 V + 1 \right)}{\mu_2^2 \left(2(1 - \lambda_2/\mu_2) \right)} \right]$$

We now find λ such that $\bar{W}$ is less than ϵ i.e. less than 4.369 milliseconds for all values of K from 1 to 9.

This has been done by using the program FIM105 written by the author and listed in appendix sixteen. For each value of K, the program starts with a large value of λ and decreases in small intervals until $\bar{W}$ is less than ϵ . However, at each step the program insures that we are considering only steady state cases by rejecting any value that would make either $\bar{W}_1$ or $\bar{W}_2$ negative.

The reader will notice here a slight imperfection in the solution. In practice, once the access arm has completed the seek it must remain in position until the record is transmitted. This implies that the arm is not free during that time for another seek. However, this condition has little impact in our analysis for two reasons. First, we have studied a system into which $\bar{W}_2$ is very small. Second, as the number of disks increases in the system the probability of consecutive accesses to the same disks is greatly reduced. This imprecision will be removed in the simulation analysis.

The values of η for K varying from one to nine are shown in table 5.1.1 and plotted in figure 5.1.2. The inter-arrival time $1/\lambda$ is graphed in figure 5.1.3.

There is little advantage in analyzing the function for more than eight disks because one must add at this point the expense of another control unit. A different type of device is probably more efficient than nine disk units.

One can easily see that the marginal arrival rate for perfect service decreases as the number of disks increases. This implies that the efficiency increases at a decreasing rate. This becomes clear by looking in figure 5.2.1 at the behavior of the inter-arrival time as the number of disks increases.

As such, there is no optimum point on the curve. The optimum configuration can be found when the marginal value of the efficiency expressed in dollars is equal to the marginal cost of the disk units. The cost of the disks drives is linear between one and eight (there are eight drives for a control unit). Therefore the marginal cost of the disks is a constant. However, the marginal value of the efficiency is not linear. It is beyond the scope of this dissertation to determine the utility or the monetary value of our measure of efficiency. However, we shall test, in the next section, the results obtained above.

ARRIVAL RATE FOR NEAR PERFECT SERVICE - EXPECTED WAITING TIME LESS THAN 4.3

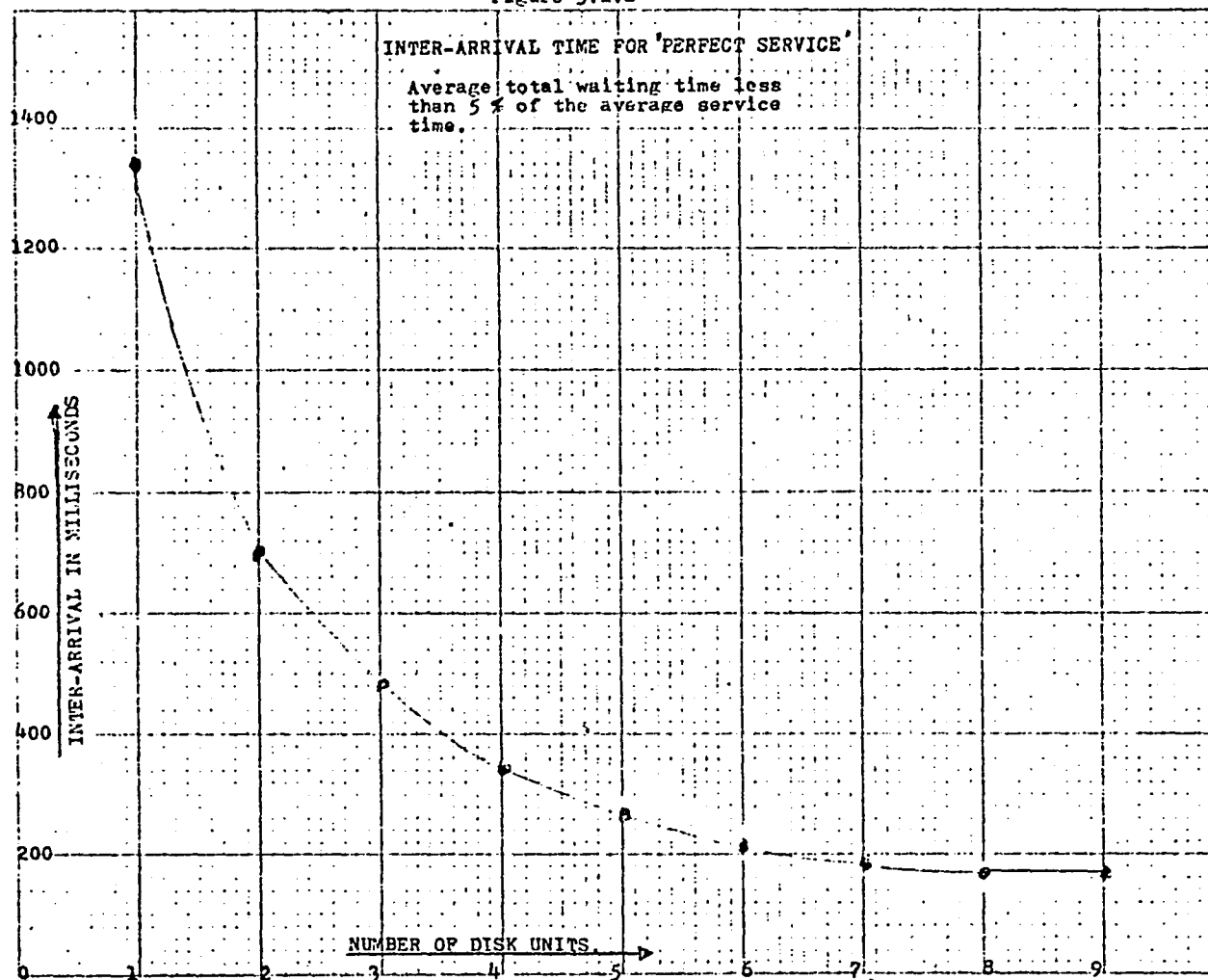
NUMBER OF DISKS K	ARRIVAL RATE λ per sec.	MARGINAL VALUE μ
1	.73464	.73464
2	1.44180	.70715
3	2.12340	.68160
4	2.78157	.65817
5	3.41418	.63261
6	4.02336	.60913
7	4.50911	.58575
8	5.17143	.56232
9	5.71245	.54102

TABLE 5.1.1

SOURCE: OUTPUT OF THE PROGRAM FIN105 LISTED
IN APPENDIX SIXTEEN

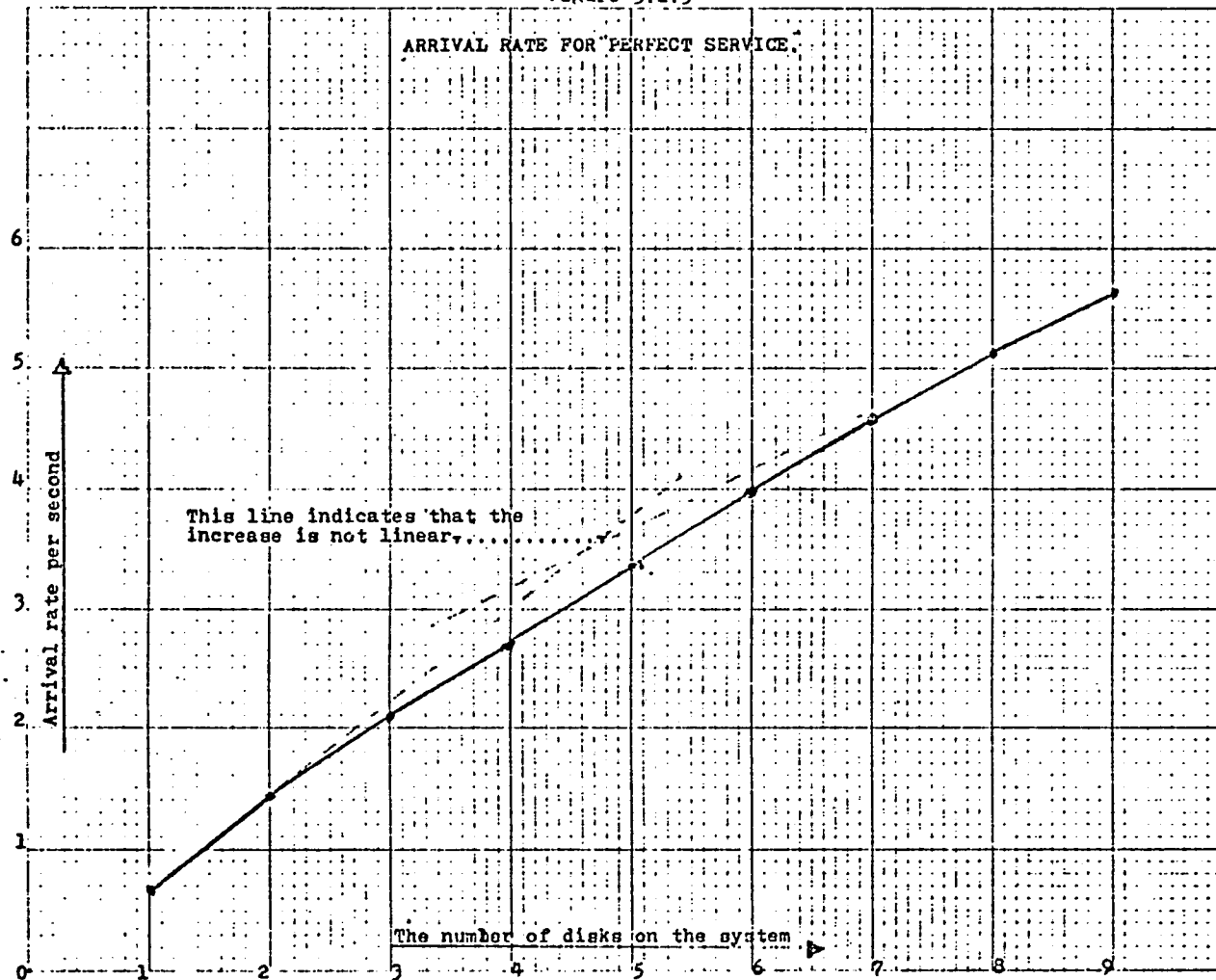
*NOTE: "4.369 milliseconds" represents 5 % of the
total expected service time.

Figure 5.1.2



SOURCE: DRAWN BY THE AUTHOR FROM THE RESULTS OF THE PROGRAM
FIM105 LISTED IN APPENDIX SIXTEEN.

Figure 5.1.3



SOURCE: DRAWN BY THE AUTHOR FROM THE RESULTS OF THE PROGRAM
FIM105 LISTED IN APPENDIX SIXTEEN.

HYPOTHESES

- A. "Near-perfect" service is maintained if the number of accesses per second varies as described above.
- B. The rate of increase of efficiency decreases as the number of disks increases.
- C. The input distribution does not have a significant effect on the response time.

SIMULATION RESULTS

The model has been tested by simulation using the program RAI0V1 written by the author and listed in appendix one. The program generates Poisson arrivals which go to any of the k disks (k varying from 0 to 9) at random. A queue is formed in front of each of the disk unit. The units then follow the model represented in figure 5.1.1. However, there is no need to use approximation procedures as in the analytical solution for the model is solved as a whole not as a sum of stages.

Testing First Hypothesis. First, we wish to verify the hypothesis that if the arrival rate follows the curve shown on figure 5.1.2 as the number of disk units increases we will have a near-perfect service.

In order to do so, the simulation was run with the arrival rate obtained in the first section. For each arrival rate and for each configuration 12 runs were made in order to obtain a confidence interval. Figure 5.2.1 and figure 5.2.2 show the changing arrival rate as the number of disks changes for a

perfect service (i.e. a negligible average waiting time) obtained from the analytical solution and used as input for the simulation runs.

Although one is interested here only in the total waiting i.e the time spent in addition to the service time, many other statistics were obtained. Tables 5.2.1 and 5.2.2 give an example of the actual output of the program. In section 1 of table 5.2.1 the program produced for each facility such statistics as the average utilization, the number of entries, and the average time of each transaction. Section 2 gives for each queue the maximum content, the average content, the number of entries, the average waiting time, and the average waiting time for the waiting units. Section 3 gives the distribution, the mean, and the standard deviation of the response time. For each rate of arrival 12 runs are made with different number of disks varying from "disk 1" to "disk 9". The waiting time shown in table 5.2.3 has been obtained by adding up for each run the expected waiting time in the two queues.

For each design, table 5.2.3 shows the average waiting times for 12 independent runs made by using as input the arrival rate obtained above. There are then 96 runs in total. In order to make inferences about the population values, the confidence intervals are shown in table 5.2.4. A graphical representation is given in figure 5.2.3. One can see that the hypothesis of perfect service is sustained. In all but two cases the expected waiting time is below the limit of 4.369 milliseconds i.e. 5% of the average service time. In the other cases, the lower 95% confidence limit is much below the 4.369 value. It indicates that we cannot reject the procedure followed by the author in the analytical solution.

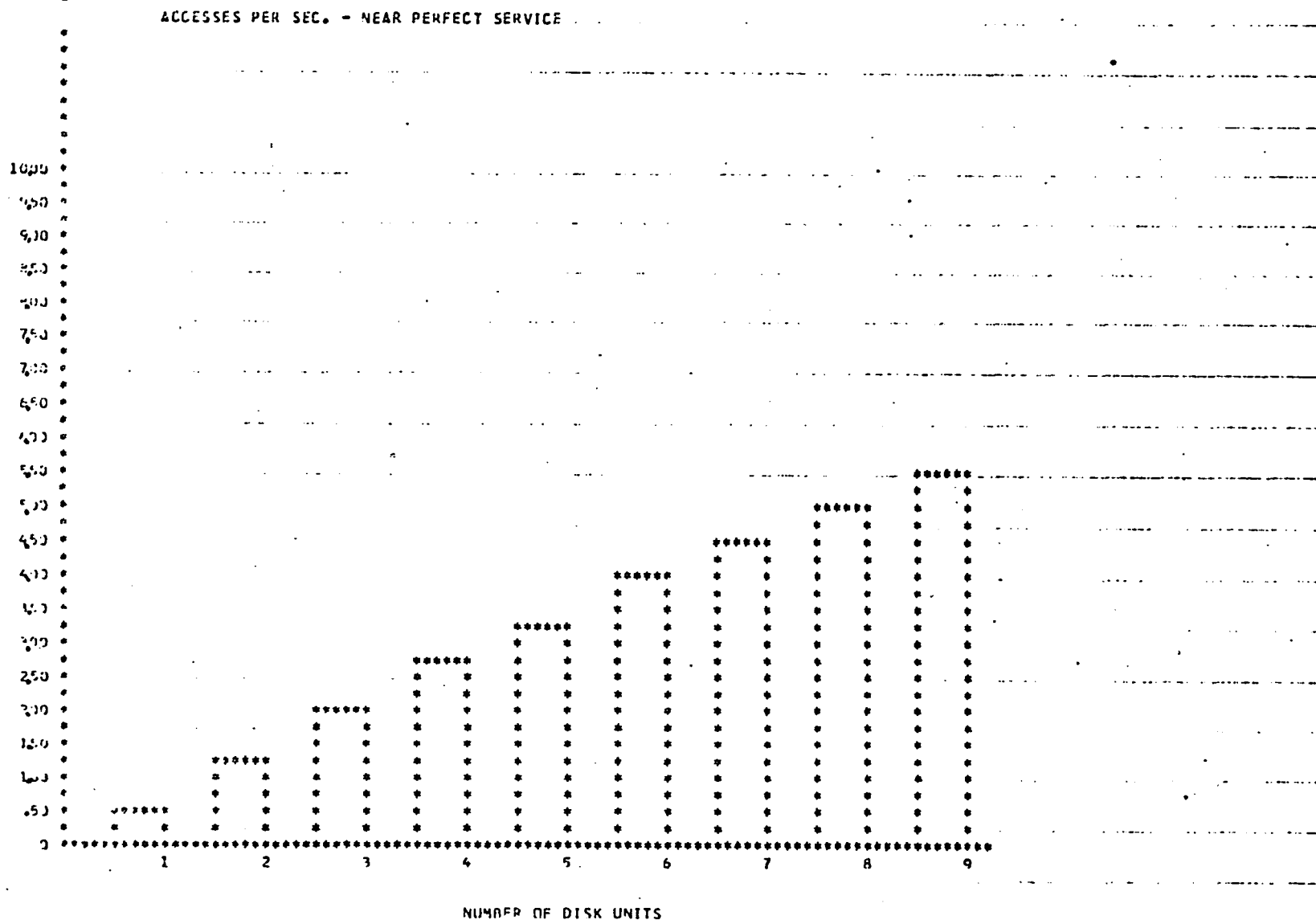


FIGURE 5.2.1

SOURCE: DRAWN FROM THE INPUT TO THE PROGRAM RAIOV1
I.E THIS IS THE INPUT TO THE SIMULATION RUNS.

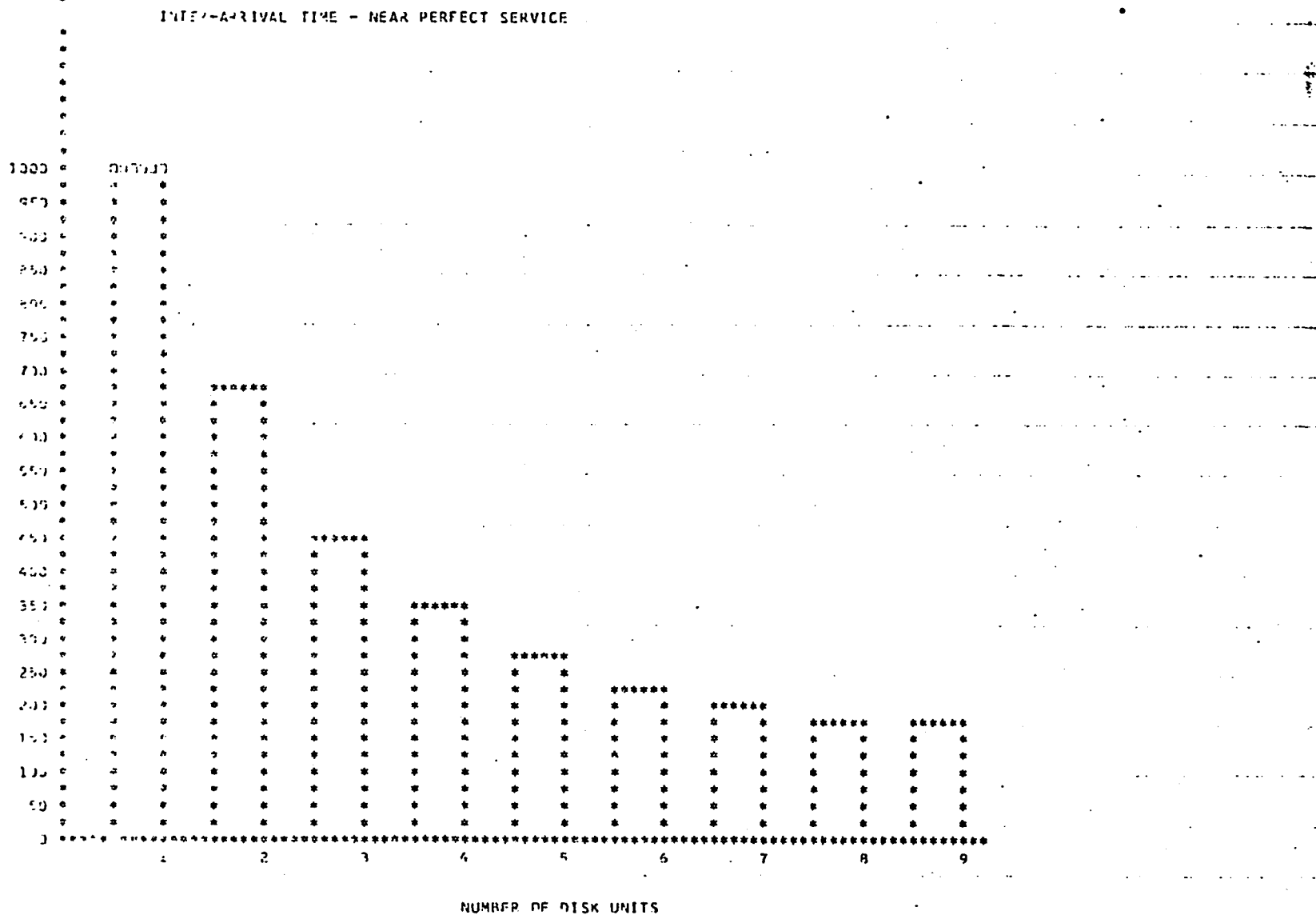


FIGURE: 5.2.2

SOURCE: DRAWN FROM THE INPUT TO THE PROGRAM RA10V1.

A GRAPH CHARACTERISTIC INDICATES THAT THE SNA VALUE IS ABOVE GRAPH LIMITS

SECTION ONE.

THE FIVE UNITS AND THE CHANNEL

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	SEIZING TRANS. NO.	PREEMPTING TRANS. NO.
01501	.006	12	117.416		
01502	.000	5	162.599		
01503	.057	11	85.000		
01504	.000	15	85.557		
01505	.006	13	117.846		
01506	.007	7	85.857		
01507	.000	13	85.923		
01508	.000	15	52.266		
01509	.068	8	97.875		
00000	.000	100	14.029		

SECTION TWO.

THE AVERAGE TIME FOR EACH UNIT

UNIT	TABLE NUMBER	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	SAVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
01501	1	.018	12	10	83.3	24.416	146.500		
01502	1	.000	5	5	100.0	.000	.000		
01503	1	.000	11	11	100.0	.000	.000		
01504	1	.000	15	15	93.7	3.125	50.000		
01505	1	.013	13	12	92.3	16.230	211.000		
01506	1	.004	7	6	85.7	9.857	59.000		
01507	1	.000	13	13	100.0	.000	.000		
01508	1	.002	15	13	86.6	2.533	19.000		
01509	1	.000	8	8	100.0	.000	.000		
00000	1	.013	100	93	92.9	.559	8.000		

SAVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

ALL WAITING I.E. TIME OVER PERFECT SERVICE

CONTENTS OF WAITING SAVED VALUES (NON-ZERO)

SAVED VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE	NR. VALUE
1	4	2	1	3	5	4	3	5	4
2	6								

SECTION THREE.

TIME SPENT IN THE SYSTEM

TABLE NUMBER	ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS	NON-WEIGHTED
100	100	100.199	94.937	10020.000	
LOWER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-1.055
100	66	65.99	65.9	34.0	-.002
200	22	21.49	87.4	12.0	1.051
300	8	7.99	95.4	4.0	2.104
400	1	.99	96.4	3.0	3.157
500	2	1.99	98.4	1.0	4.211
600	1	.99	100.0	.0	5.264

REMAINING FREQUENCIES ARE ALL ZERO

TABLE: 5.2.2

SECTION ONE.

THE LIFE OF THE AND THE CHANNEL

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN	SIZING TRANS. NO.	PREEMPTING TRANS. NO.
G1501	.079	20	89.949		
G1502	.043	26	82.153		
G1503	.146	16	74.562		
G1504	.057	13	81.722		
G1505	.065	21	80.095	1	
G1506	.147	100	12.279		

SECTION TWO.

THE WAITING TIME FOR EACH UNIT

DATE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	SAVEPAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
01/29	1	.000	20	20	100.0	.100	.100		
01/29	2	.007	25	22	87.9	7.279	60.666		
01/29	1	.003	16	15	93.7	6.062	97.000		
01/29	1	.001	18	16	88.9	2.611	23.500		
01/29	1	.002	21	20	95.2	3.190	67.000		
01/29	1	.001	100	97	96.9	.409	13.666		

120 1404 TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

ALL NOTING I.E. TIME OVER PERFECT SERVICE

TABLE 1. SELF-REPORTED SAVELIVES (1991-2000)

NR, VALUE	NR, VALUE	NR, VALUE	NR, VALUE	NR, VALUE	NR, VALUE	NR, VALUE	NR, VALUE
1	2	3	4	5	6	7	8

SECTION THREE.

TIME SPENT IN THE SYSTEM

TABLE TITLE		MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS		WGT-WEIGHTED
FURTHER TITLE		86.979	66.052	8699.000		
ORDER	RESERVED	PER CENT	CUMULATIVE	CUMULATIVE	MULTIPLE	DEVIATION
LIMIT	FREQUENCY	OF TOTAL	PERCENTAGE	REMAINDER	OF MEAN	FROM MEAN
0	0	.00	.0	100.0	-0.000	-1.315
100	69	68.99	68.9	31.0	1.151	.199
200	24	23.99	92.9	7.0	2.302	1.712
300	5	4.99	97.9	2.0	3.453	3.226
400	2	1.99	100.0	.0	4.604	4.739

REMAINING FREQUENCIES ARE ALL ZERO

REMAINING FREQUENCIES ARE ALL ZERO

TABLE: 5.2.1

SOURCE: SAMPLE OUTPUT OF THE PROGRAM RALOV1 LISTED
IN APPENDIX ONE.

EXPECTED TOTAL WAITING TIME
FOR 96 RUNS
USING THE ARRIVAL RATE OBTAINED FROM THE ANALYTICAL SOLUTION

NUMBER OF DISKS	ARRIVAL RATE PER SEC	INTER ARRIVAL TIME IN MS.	WAITING TIME AT ALL STAGES FOR DIFFERENT RANDOM NUMBERS MEANS OF RUNS		AVERAGE
1	0.73	1339	13 05 03 03 01 07 04 04 04 02 04 02	4.333	
2	1.441	693	05 12 02 04 00 01 01 01 02 02 03 06	3.250	
3	2.123	470	07 04 02 01 09 06 03 05 01 03 05 03	4.093	
4	2.781	359	06 05 03 03 09 02 03 04 01 04 03 05	4.000	
5	3.414	292	03 02 05 05 01 01 09 04 03 03 03 03	3.500	
6	4.023	248	08 04 01 03 10 02 05 03 04 02 01 07	4.583	
7	4.609	217	12 11 02 03 17 02 03 03 09 05 04 00	5.916	
8	5.171	193	00 02 04 05 01 11 04 03 01 02 05 00	3.166	
9	5.712	175	00 05 01 01 06 01 06 04 01 03 01 05	3.25	

TABLE: 5.2.3

SOURCE: SUMMARY OF RESULTS OBTAINED BY 12 RUNS PER DISK
OF THE PROGRAM RAIOV1 LISTED IN APPENDIX ONE.

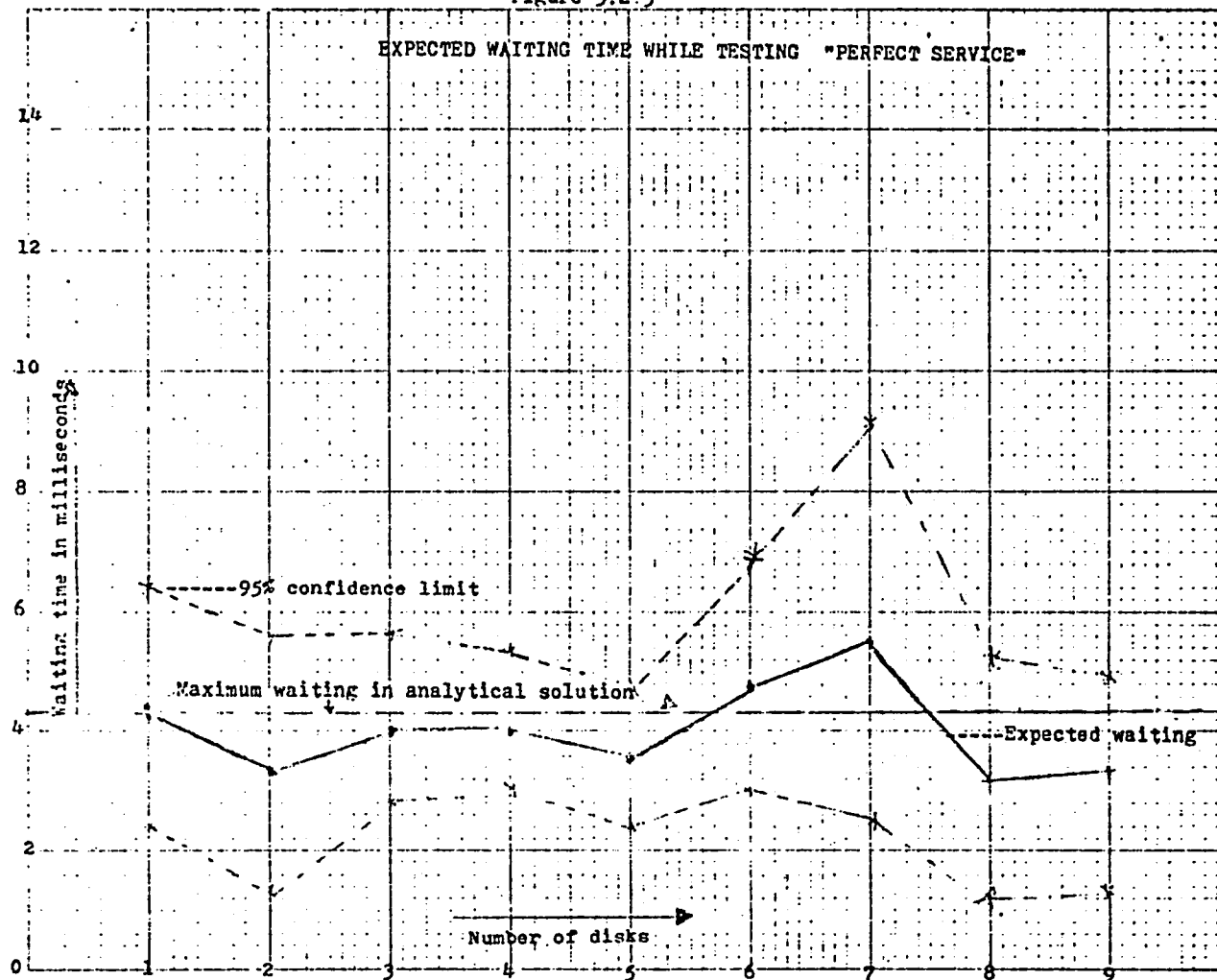
CONFIDENCE LIMITS FOR AVERAGE WAITING TIME

NUMBER OF DISKS	ARRIVAL RATE PER SEC.	INTER ARRIVAL TIME IN MS.	TOTAL WAITING TIME			CONFIDENCE INTERVAL FOR POP. MEANS	
			ANAL.	SIM.	STD. ERR. OF MEANS	95%	99%
1	0.734	1339	< 4.369	4.333	0.90	2.339 - 6.328	1.526 - 7.141
2	1.441	693	< 4.369	3.250	0.946	1.169 - 5.330	0.321 - 6.178
3	2.123	470	< 4.369	4.083	0.701	2.541 - 5.625	1.913 - 6.253
4	2.781	359	< 4.369	4.000	0.603	2.674 - 5.325	1.355 - 6.643
5	3.414	292	< 4.369	3.500	0.621	2.133 - 4.866	1.576 - 5.423
6	4.023	248	< 4.369	4.583	0.874	2.560 - 6.505	1.877 - 7.288
7	4.609	217	< 4.369	5.717	1.484	2.652 - 9.180	1.323 - 10.510
8	5.171	193	< 4.369	3.166	0.877	1.237 - 5.096	0.451 - 5.892
9	5.712	175	< 4.369	3.250	0.779	1.535 - 4.964	0.837 - 5.662

TABLE: 5.2.4

SOURCE: PREPARED BY THE AUTHOR FROM THE RESULTS SHOWN IN TABLE 5.2.3.
 THE SIMULATION RESULTS ARE FROM THE PROGRAM RAJOVI LISTED IN APPENDIX ONE.
 THE CONFIDENCE LIMITS HAVE BEEN PRODUCED BY THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR
 AND LISTED IN APPENDIX SEVEN.

Figure 5.2.3



SOURCE: DRAWN FROM THE RESULTS SHOWN IN TABLE 5.2.4

74

Testing the Second Hypothesis. In the previous section we have shown that the simulation runs have given the same results as the analytical solution. The efficiency, measured as the maximum arrival rate for near-perfect service, increases at a decreasing rate as obtained from the analytical solution. However, the usual criterion in such a system is not the ability to have a negligible waiting time but the length of the response time. In order to study the response time, the model was run under an arrival rate of 5 units per second. The results are summarized in the first column of table 5.2.5 and represented for a poisson assumption in figures 5.2.4 and 5.2.5. A confidence interval is also provided in order to make an inference about the population.

One can obtain for each configuration the average utilization of the disk by adding the utilization figures and by dividing this total by the number of disks. Figure 5.2.4 shows how the utilization decreases by going from one to nine disks. It is interesting to note that there is a significant decrease in utilization by going from one to two drives as it was mathematically determined.

As we have concluded in the analytical solution, the efficiency measured by the possible arrival rate for perfect service or by the expected total response time is increasing at a decreasing rate as the number of disks increases from 1 to 9. The optimum configuration can be obtained only by knowing the opportunity cost of delay.

EXPECTED RESPONSE TIME
OBTAINED FOR THREE ARRIVAL DISTRIBUTIONS

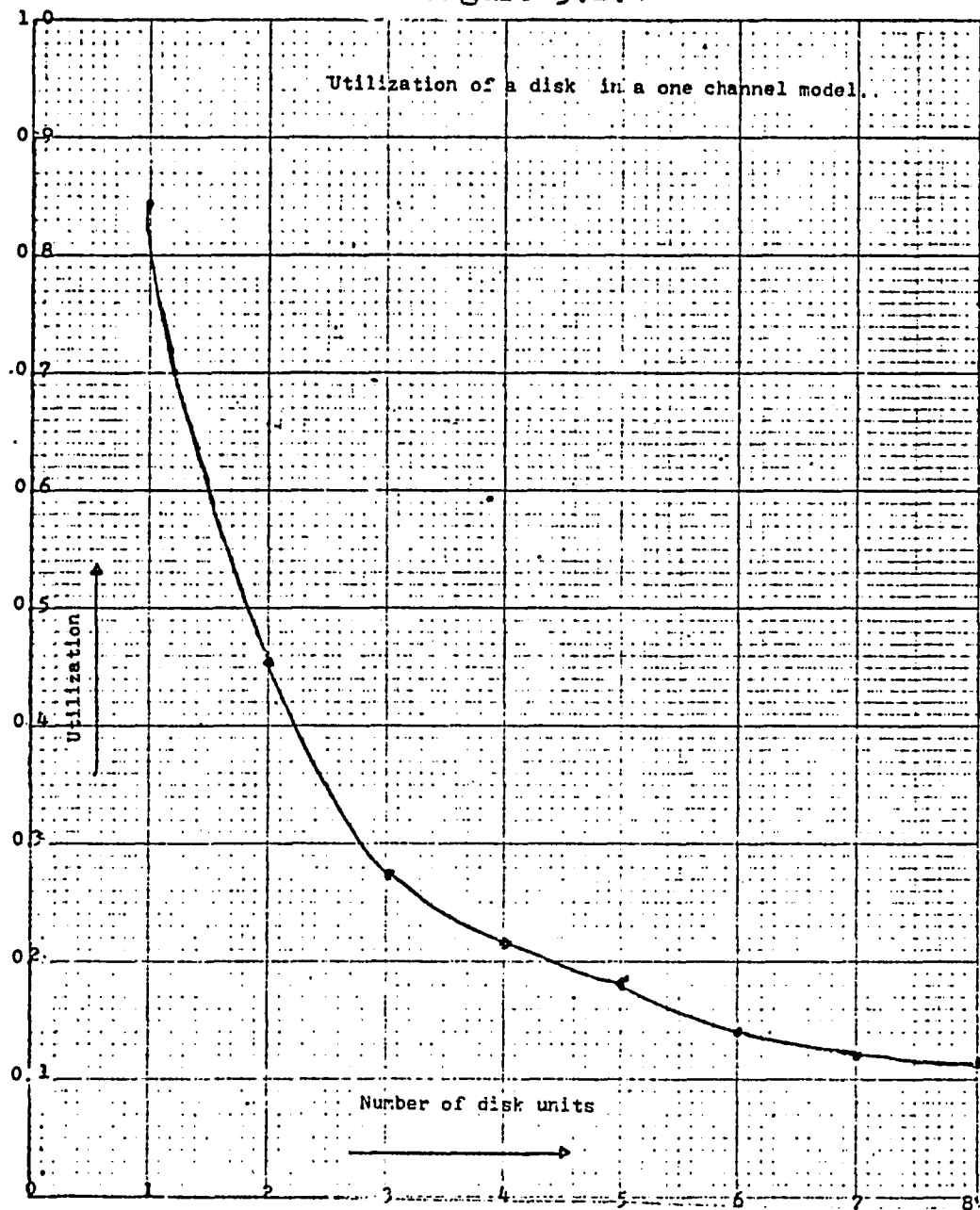
NUMBER OF DISKS	ARRIVAL DISTRIBUTION											
	POISSON				UNIFORM				NORMAL			
1	114	144	129	122	114	100	113	104	091	88	86	83
	139	121	144	140	90	96	96	100	87	87	90	83
	122	118	133	121	102	101	109	108	88	90	90	90
2	104	102	97	93	99	98	94	92	88	85	93	88
	99	92	101	99	96	100	96	95	85	86	91	87
	103	98	104	102	95	97	90	98	88	87	85	90
3	101	097	99	100	89	90	88	91	87	90	92	90
	95	95	93	98	97	95	92	95	89	89	87	89
	98	95	99	97	99	96	93	96	82	84	95	87
4	100	90	100	093	93	84	91	100	85	89	90	85
	94	96	97	92	95	95	99	94	87	86	86	85
	96	99	96	95	88	91	93	92	85	88	87	89
5	95	97	91	89	93	86	86	83	082	87	90	85
	95	93	97	95	88	90	93	89	89	86	88	87
	98	93	96	97	90	95	94	90	86	90	92	85
6	92	94	95	95	89	91	86	85	90	87	88	87
	88	89	95	91	88	88	92	90	87	92	93	87
	91	92	93	92	96	87	90	89	92	90	84	88
7	92	87	96	90	90	88	95	87	87	87	88	86
	96	93	94	90	85	90	95	90	84	97	89	89
	92	91	93	90	93	92	87	93	92	85	88	92
8	92	91	90	92	91	92	94	92	92	83	87	87
	91	91	97	83	92	91	93	92	83	87	84	90
	89	92	93	89	96	83	91	93	83	86	90	86
9	84	95	93	94	94	97	83	86	90	87	93	97
	87	87	87	89	94	90	91	90	91	91	83	88
	91	90	91	91	91	89	88	86	85	86	90	87

NOTE: EACH VALUE SHOWN IN THIS TABLE IS THE AVERAGE OF ONE RUN MADE OF
 THERE 5 ARRIVALS PER SECOND.
 THE RESPONSE TIME IS EXPRESSED IN MILLISECONDS.
 100 NON TRANSIENT OBSERVATIONS.
 12 DIFFERENT SEQUENCES OF RANDOM NUMBERS WERE USED.
 IN EACH OF THE 324 RUNS THE SYSTEM WAS FIRST LOADED WITH 50 ARRIVAL

TABLE: 5.2.5

SOURCE: SUMMARY OF THE RESPONSE TIME OBTAINED BY 324 RUNS OF THE PROGRAM
 RAIOVI LISTED IN APPENDIX ONE.

Figure 5.2.4

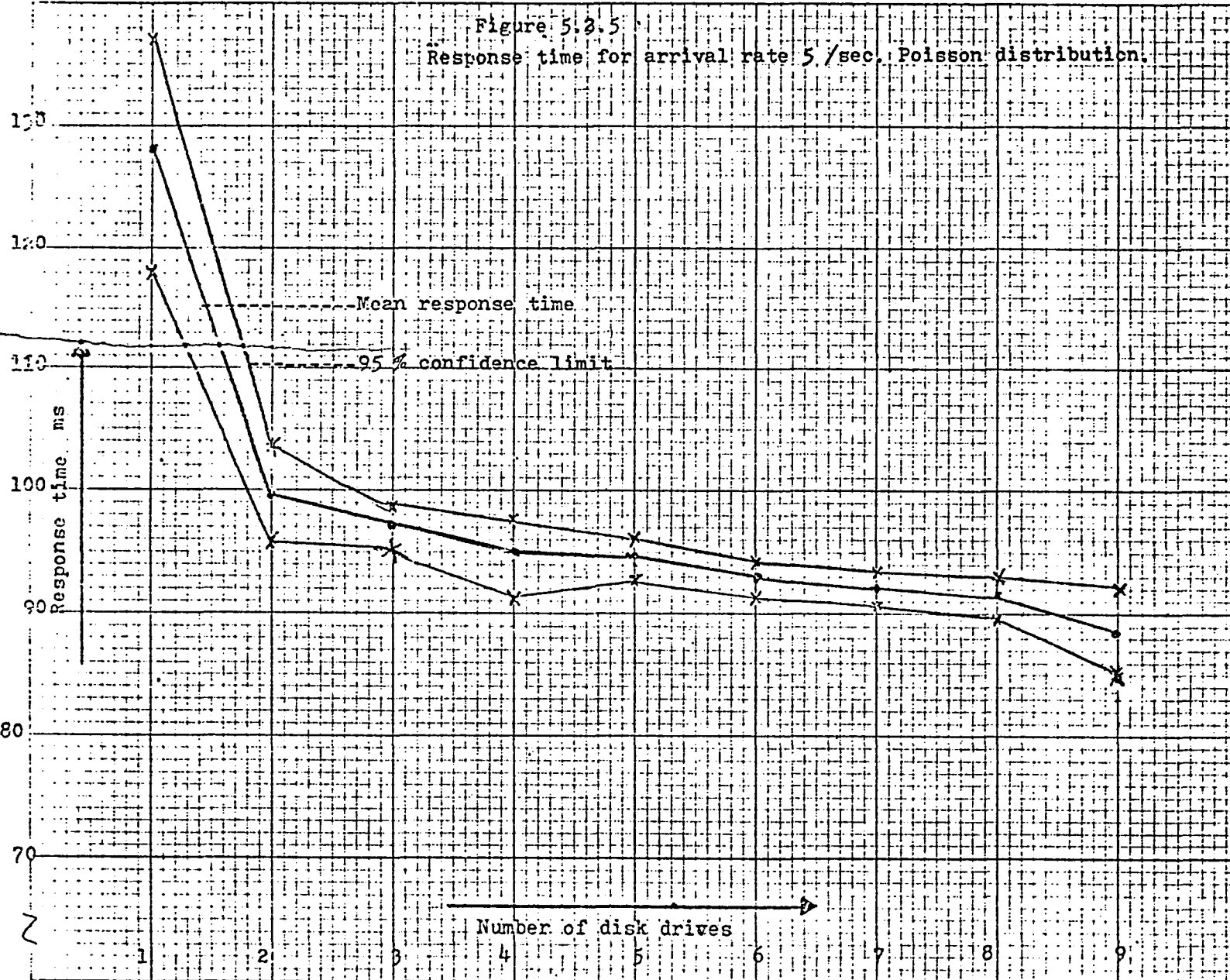


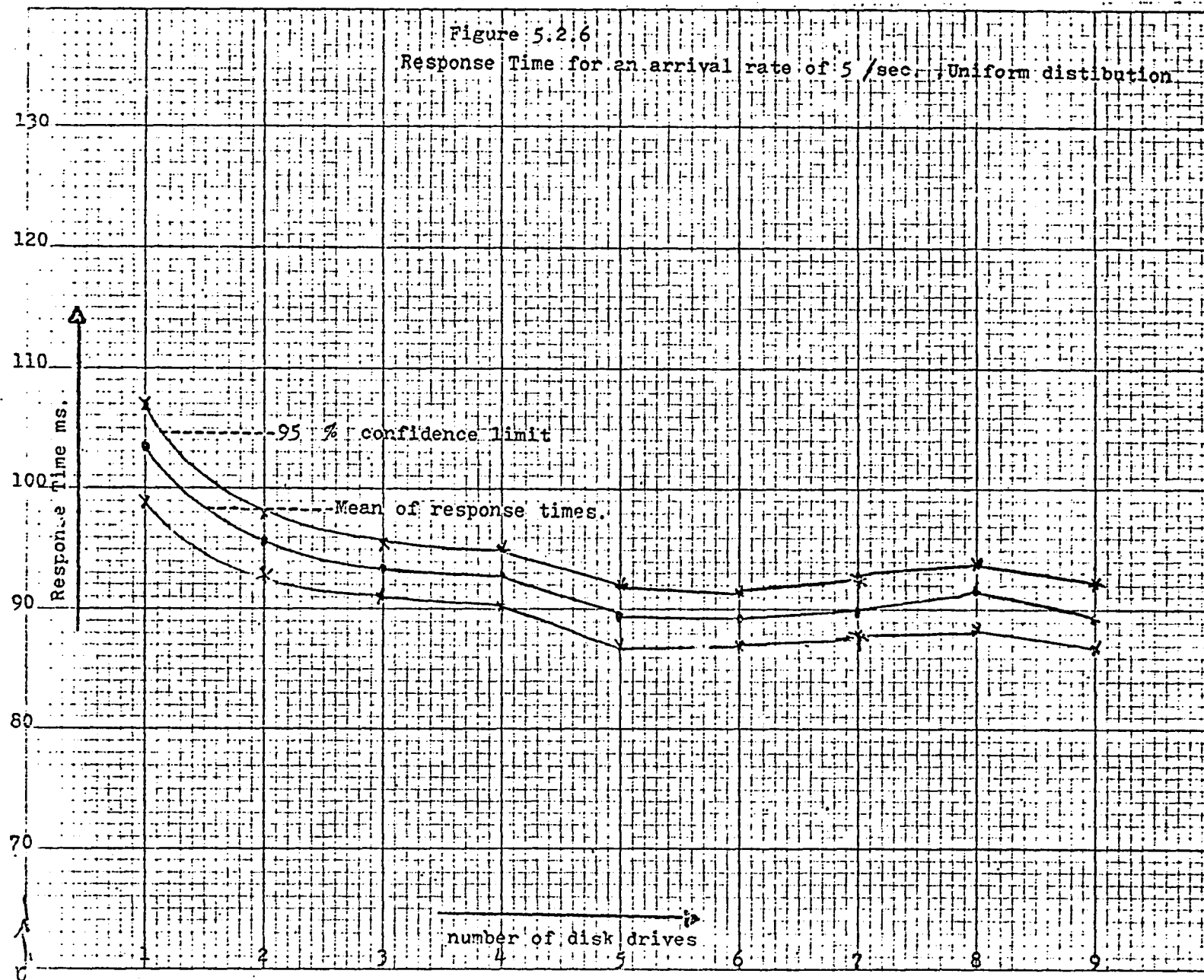
Drawn according to the results provided
by many runs of the program RAI0V1

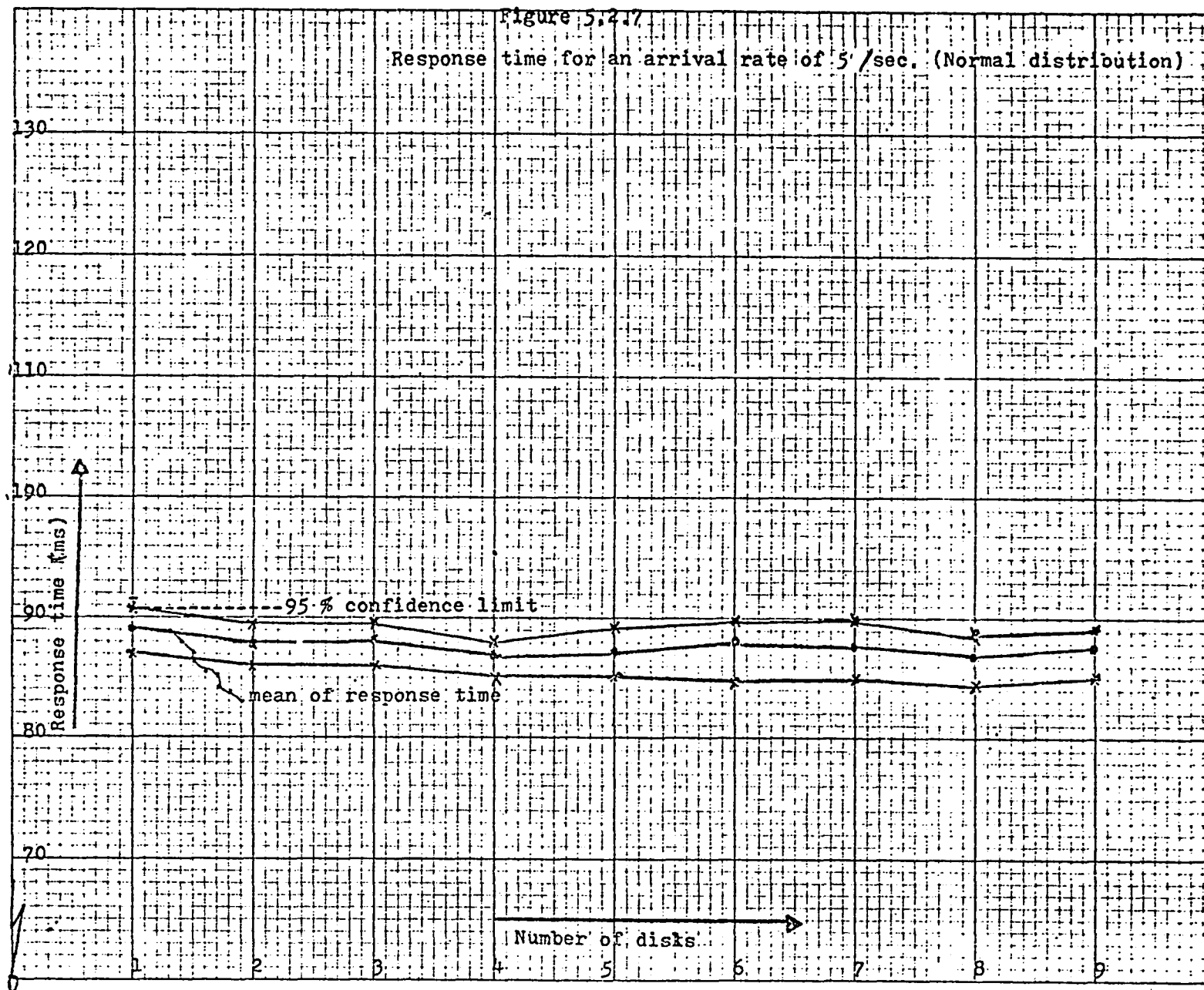
Testing the Third Hypothesis. In order to test the effect of the arrival distribution on the total response time, 12 runs were made for each of 9 disks and for three different distributions i.e. Poisson, Normal and Uniform. The means of all the distributions of the inter-arrival time are the same i.e. 200 milliseconds. The standard deviation of the normal distribution is 40. In the uniform distribution, the inter-arrival times varies from 0 to 400 milliseconds. An analysis of variance has been performed on the results. The output of the analysis of variance is shown on table 5.2.6 and table 5.2.7. In table 5.2.6, all the results were included i.e. for the nine disks. In table 5.2.7, the values for the first and the last disk had been removed because in practice only eight disks are supported by a single control units as discussed in chapter two. If there is only one disk unit we have a different model i.e. a series of two single servers.

A graph of the response time and of the confidence intervals is given for each arrival distribution in figures 5.2.5, 5.2.6, and 5.2.7. One can easily see that, unlike in the previous model studied in chapter four, the arrival pattern has a significant effect on the expected response time. Furthermore, there is a significant interaction factor. It implies that the number of disks on the system together with the arrival pattern will have a significant effect on the response time. The third hypothesis is rejected.

140







EFFECT OF ARRIVAL PATTERN ON THE RESPONSE TIME

ANOVA--LEAST SQUARES SOLUTION				
SOURCE	SS	DF	MS	F
A	8657.000	8.	1082.125	90.865
S(A)	1179.000	99.	11.909	
B	5413.000	2.	2706.500	165.295
AB	7158.000	16.	447.375	27.604
BS(A)	3209.000	198.	16.207	
GRAND MEAN = 92.923				
ANOVA--UNWEIGHTED SOLUTION				
SOURCE	SS	DF	MS	F
A	9661.250	8.	1082.531	90.900
S(A)	1179.000	99.	11.909	
B	5412.750	2.	2706.375	165.997
AB	7161.250	16.	447.516	27.612
BS(A)	3209.000	198.	16.207	
GRAND MEAN = 92.923				
CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B				
	1	2	3	
1	123.917	103.333	88.167	
2	99.500	95.917	88.167	
3	97.250	93.417	89.250	
4	94.917	92.917	86.933	
5	94.667	90.167	87.333	
6	92.333	89.250	88.750	
7	92.000	90.500	87.833	
8	91.250	92.667	86.500	
9	89.917	89.917	87.833	

NOTE: A = NUMBER OF DISKS ; B = ARRIVAL PATTERN
A AND B ARE FIXED FACTORS

TABLE: 5.2.6

SOURCE: OUTPUT OF THE PROGRAM FINADV LISTED IN APPENDIX FIVE

EFFECT OF ARRIVAL PATTERN ON THE RESPONSE TIME

ANOVA--LEAST SQUARES SOLUTION				ANOVA--UNWEIGHTED SOLUTION			
SOURCE	SS	DF	MS	SOURCE	SS	DF	MS
A	694.000	6		A	696.000	6	
S(A)	731.000	77		S(A)	731.100	77	
B	1955.000	2		B	1954.500	2	
AB	401.000	12		AB	403.500	12	
RS(A)	1547.000	154		RS(A)	1547.000	154	
GRAND MEAN = 91.468				GRAND MEAN = 91.468			
12.184	115.667	9.494	97.308	12.219	115.666	9.494	97.283
F			3.327	F			3.347

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

1	91.251	92.633	86.500
2	90.900	95.917	89.167
3	97.250	93.417	89.250
4	94.917	97.917	96.333
5	94.667	91.167	87.333
6	92.533	89.250	89.750
7	92.000	91.500	87.933

NOTE: A = NUMBER OF DISKS : 3 = ARRIVAL PATTERN
A AND B ARE FIXED FACTORS

TABLE: 5.2.7

SOURCE: OUTPUT OF THE PROGRAM SIMADY LISTED IN APPENDIX FIVE

INFERENCE FROM A SAMPLE 02/04/72

VALUE OF SAMPLE STATISTICS

LIST OF THE INPUT DATA

92.00	94.00	95.00	95.00	95.00	88.00
89.00	95.00	91.00	92.00	93.00	92.00

SIZE OF THE SAMPLE :	12
SAMPLE MEAN VALUE :	92.5833
VARIANCE OF SAMPLE :	5.2414
SAMPLE STD DEVIATION:	2.2894
EST POP STD DEV :	2.3912
STANDARD ERROR MEANS	.6903

CONFIDENCE LIMITS ON POPULATION MEAN:

CONF LEVEL	LOWER LIM	UPPER LIM
50.000000	92.101898	93.064743
75.000000	91.745102	93.421539
90.000000	91.344269	93.822372
95.000000	91.065704	94.100937
99.000000	90.447510	94.719131
99.899994	89.556870	95.609772
99.989990	88.602249	96.564392
99.998993	87.598679	97.567963

Table: 5.2.8

Source: A sample output of the program FINCON01 written by the author and listed in appendix seven.

CONCLUSION

In this chapter, simulation has been used to test the analytical solution of a direct access method model which is of major importance in any real-time or time-sharing system. The first hypothesis concerning the maximum rate of arrival for perfect service was not rejected. We did not have enough evidence to reject the second hypothesis.

The third hypothesis concerning the effect of the arrival pattern on the response time has been rejected. It implies that, unlike the real-time model tested in Chapter Four, this model is sensitive to the arrival distribution and to the interaction of the arrival distribution and the number of disks. It implies that it is worthwhile to make sure of the arrival distribution by field tests.

If the real arrival distribution exhibits some departure from the classical "Poisson" distribution, it is not possible to use the same formulae to get the response time. However, the simulation program can easily handle any arrival distribution, and can be used for more realistic service distribution.

In the previous model, studied in Chapter Four, the arrival pattern had no effect on the system with 20 terminals at the first stage. In this chapter, we have found that the interaction of the number of servers in the first stage and the arrival pattern was significant. This suggests that as the number of disks increases the system may become less sensitive to the arrival pattern and therefore less sensitive to a departure from the classical Poisson arrival.

Since the model considered in this chapter is in fact only a submodel of a real-time model, it is possible to integrate it into a larger model. Computer simulation makes it possible to study the interaction of subsystems.

CHAPTER SIX

TIME-SHARING MODELS

The purpose of this chapter is to test by simulation the analytical solutions obtained for some time-sharing models with different queue disciplines and to test the sensitivity of the model to a departure from the usual "Poisson" assumption. Several tests will be made for all queue disciplines so that comparisons can be made between them. The basic characteristic of a time-sharing system is the sharing of the services of a computer among a number of independent users who are linked with the computer from remote locations. Each user shares the available processing time with all the other users. Each user is served in turn by the computer for a time period that is called a quantum. Depending on the system, the quantum may be a fixed or variable length of time. The objective in designing a time-sharing system is to develop a servicing discipline which provides an effective allocation of the computer facilities.

There are two important criteria of efficiency of a time-sharing system. First, the congestion expresses the number of users in the system at any given time. Second, the response time which is the total time spent in the system. The response time is divided into the waiting time and the service time. In this chapter we shall use the waiting time as our measure of efficiency. Congestion is a measure useful to computer center managers. However, users are not interested in congestion but in the response time or in the waiting time..

The first analytical model of time-sharing was developed by Coffman and Krishnamoorthi¹. Assuming a geometric service time distribution and later an exponential distribution they arrived at the stationary distribution of the number of users in the system and the expressions for the first two moments. Later, Krishnamoorthi and Wood² expanded the work of the above authors and derived the exact distribution of the number in the queue at the end of the quantum. They also obtained the conditional response time i.e. dependent on the length of service required, in a Round-Robin Model with a fixed number of inputs. Chang³ found the transient and the stationary queue distribution for a simplified time-sharing model where the quantum, instead of being fixed, is a random variable directly related to the job's processing need. An unrealistic assumption that each job is a multiple of the quantum of service is made to simplify the solution. Kleinrock⁴, assuming an infinite source population, derived the expected total time in the queue conditional on the service requirement in a Round Robin model. In a subsequent paper Coffman and Kleinrock⁵ expanded the work

¹E. G. Coffman and B. Krishnamoorthi, Preliminary analyses of Time-Shared Computer Operation, Doc. SP-1719, (Santa Monica, California: System Development Corp., August 1964)

²B. Krishnamoorthi and R. C. Wood, "Time-Shared Computer Operations with Both Inter-Arrival and Service Times Exponential", Journal of ACM, Vol. 13, No. 3, (July 1966), p. 317-318.

³W. Chang, "A Queuing Model for a Simple Case of Time-Sharing," IBM System Journal, Vol. 5, No. 2, (1966) p. 115-124

⁴L. Kleinrock, "Time-Shared Systems, a Theoretical Treatment," Journal of ACM, Vol. 14, No. 2, (April 1967) p. 242-261

⁵Edward Coffman, Leonard Kleinrock, "Feedback Queuing Model for Time-Shared Systems," Journal of ACM, Vol. 15, No. 4 (October 1968), p. 549-576

of Kleinrock and derived the exact expression for the expected waiting time conditional on the service requirement assuming a zero set-up time. The different queue disciplines that they have studied, such as Round Robin, many levels of feedback, and 'shortest job first', will be analyzed in detail in this chapter. Adiri and Avi-Itzhak⁶ have studied models where there are a limited number of feedbacks and a limited number of sources. They derived the expected response time conditional on the service requirement, something Krishnamoorthi and Wood were unable to do. They indicated that in the finite model⁴ queue length distribution is the same as the machine interference problem. Rash⁷, including the set-up time, compared the different scheduling algorithms in term of costs of delay. He indicated the need to use the classical Poisson input process for ease of the solution and to enable

comparison with others. However, he expressed the urgent need to develop a solution based on more realistic non-Poisson assumptions. This is what we shall do in this chapter.

Recently Heacox and Purdom⁸ have developed a modified

6

Adiri and B. Avi-Itzhak, "A Time-Sharing Model with Many Queues," Operations Research, Vol. 17, No. 6, (November 1969) pp. 1077-1089. I. Adiri, and B. Avi-Itzhak, "A Time-Sharing Queue," Management Science, Vol. 15, No. 11, (July 1969), pp. 639-657. I. Adiri, and Avi-Itzhak, "A Time-Sharing Queue with a Finite Number of Customers," Journal of the ACM, Vol. 16, No. 2, (April 1969) pp. 315-323.

7

Philips Rash. "A Queuing Theory Study of Round Robin Scheduling Time-Shared Computer Systems," Journal of ACM, Vol. 17, No. 1, January 1970, pp. 131-135

8

Harry C. Heacox and Paul W. Purdom. "Analysis of Two Time-Sharing Queuing Models." Journal of ACM, Vol. 19, No. 1, January 1972. pp. 70-91.

Round-Robin with overhead, in which the amount of service per pass depends on the arrival rate. The system is studied in terms of the mean cost due to the waiting time in queue. The results are somewhat different than the ones proposed by Rash who also used the cost of delay as a criterion of efficiency.

In this chapter, we will review the analytical results in terms of expected waiting time for several scheduling algorithms and see whether the same relationship holds for the simulation. Furthermore, following the wish expressed by many authors, we will develop solutions for non-Poisson arrivals and see whether the arrival distribution has any influence on the overall response time.

MODELS TO BE STUDIED

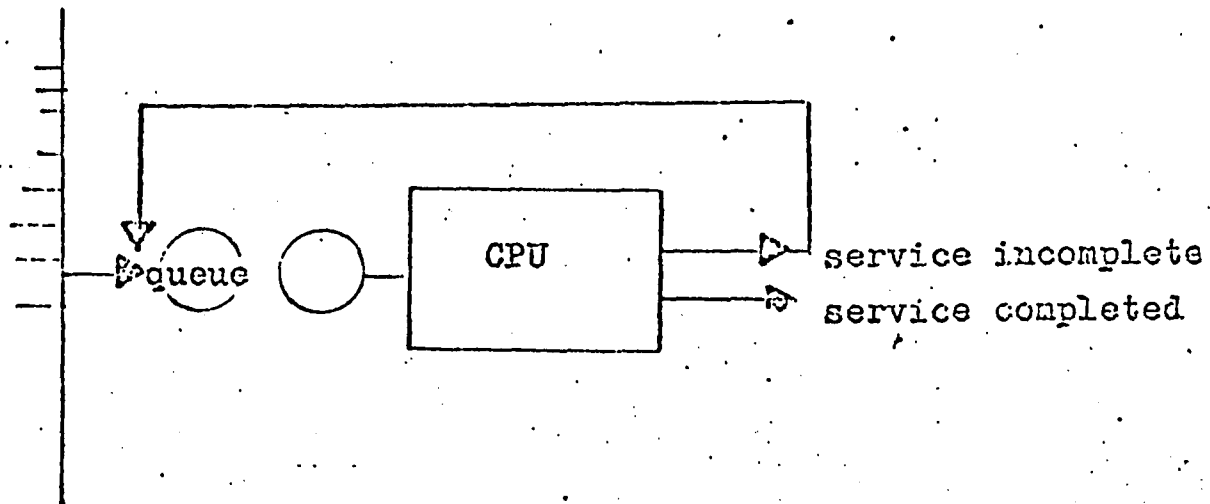
In this chapter three different scheduling techniques will be analyzed. Usually, in a time sharing system the user is given a time slice of service called "quantum". If the service is completed within a quantum, the customer goes out of the system. If the service is not completed the unit goes back at the end of the waiting line and waits for another quantum. In the Round-Robin model the unit will go at the end of the queue with arriving units. In the feedback model, also referred to as Foreground-Background model, the unit will go to the end of a lower queue. If a unit requires 3 quanta it will use one quantum, then go to the end of the queue number 2. When queue number 1 is empty, the system will accept units from the second queue. The unit will be serviced, and if it needs a third quantum it will go to the end of queue 3. The system that is valid for n levels will be referred to as FB n . If n is not limited it will be referred to as FB ∞ .

We referred to the regular "first come first served" as FSFC and to SJF for the queuing discipline where there is advance knowledge of length of service required and where the shortest job goes first.

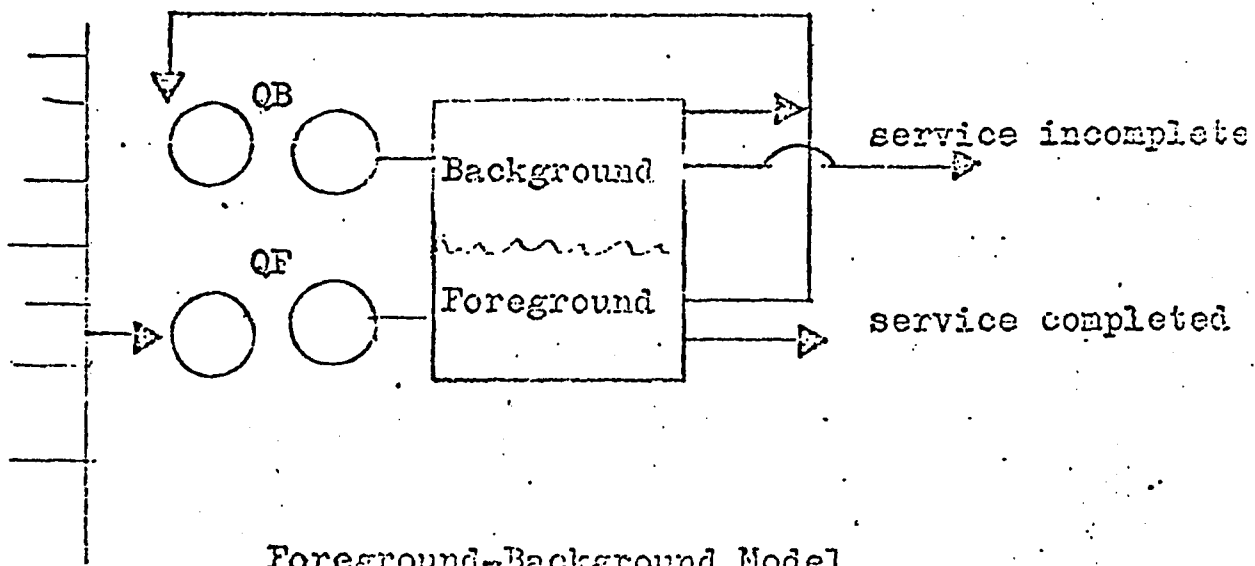
Figure 6.1.1 shows the Round-Robin model where the unit interrupted by the end of the quantum goes back to the end of the unique queue while in the FB model the interrupted unit goes to the end of a second queue. Figure 6.1.2 shows a FB model where there are many levels of queues. If service is not completed after the first pass, the unit goes to the end of the second queue. If service is not completed after the second time, it goes to the end of the third queue and so on. In the priority case, some unit having low priority may join the k^{th} queue or, after the k^{th} pass, a unit may go on to a higher queue.

PROCEDURE

Before one can study in detail the different time-sharing models, one must determine the procedure to be followed. In this chapter, we will describe each model in detail, present an analytical solution, do simulation experiments, and compare the analytical results with the simulation results and see whether the relationships among the queue disciplines do hold as expressed in the analytical solution. However, in the design of an experiment one must also consider the cost of the experiment, i.e. one must get as much reliable information as possible at a minimum cost.



Round-Robin Model



Foreground-Background Model

Figure 6.1.1 Two Models For Time-Sharing

Source: J. M. McKinney, "A Survey of Analytical Time-Sharing Models", Computing Survey, Vol. 1, No. 2, (June 1969) p. 106.

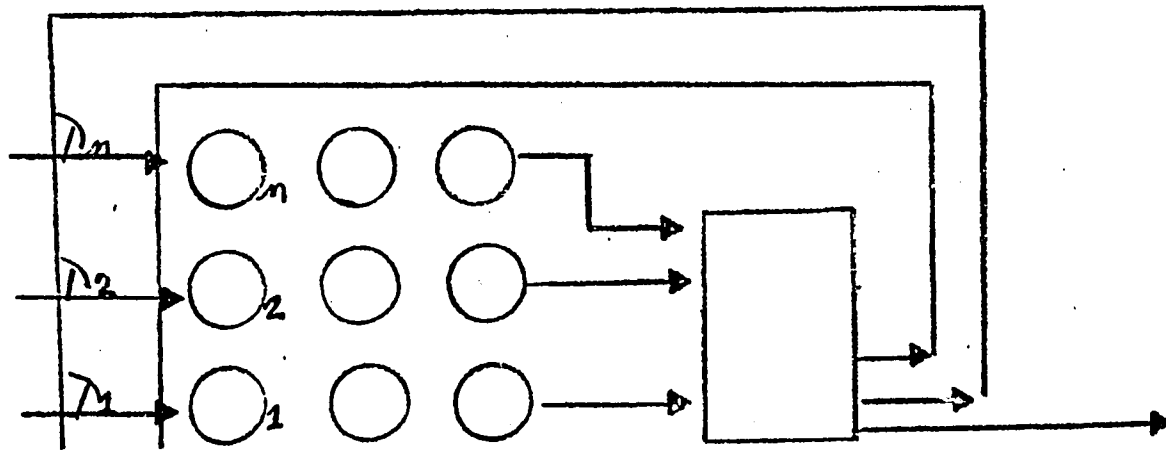
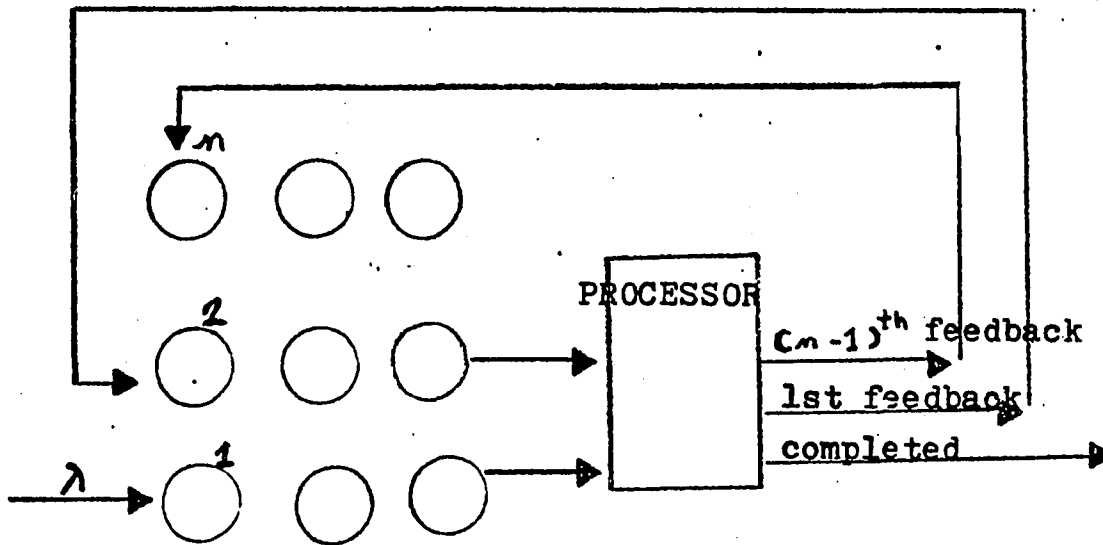


Figure 6.1.2 - The FB model with n levels of queues.

(upper without priority; lower case with priority.)

Source: Modification of the figures presented in:

E. Coffman and L. Kleinrock, "Feedback Queuing Model for Time Shared Systems", Journal of ACM, Vol. 15, No. 4. (October 1968), p. 554.

In studying and testing by simulation these time-sharing models one is faced with a factorial design involving four factors; the queue discipline, the utilization of the CPU, the quantum size, and the arrival distribution. There are four queue disciplines, three arrival distributions, seven levels of CPU utilization, and five quantum sizes. There are 420 cells and 12 mean values per cell. We then have 5040 runs to do. If each run has 100 non transient observations there will be 504,000 units to measure excluding the units necessary to load the system each time.

Simulation experiments are expensive. Therefore one should try to reduce the number of runs required. If one is willing to settle for less than a complete investigation a fractional factorial design can be used. The author has used a different approach. As in a response surface design preliminary experiments were conducted to determine the areas which should be investigated further.

We will examine the sensitivity of only some of the queue disciplines and also make comparisons between the models. We will now study the analytical solution, describe simulation experiments, and compare separately the results of the Round Robin model, Feedback model, and the Shortest Job First model. The classical FIFO model will be used as reference. In order to be able to make any comparison between the discipline we will compare all of them in terms of the expected total time which will be the expected total time spent in the system less the service time.

ROUND-ROBIN SCHEDULING

The Model. In a simple Round-Robin model, the users enter and join a single queue in a first-in first-out manner. The user is given a limited service time called "quantum". If the service is completed within that small service time it goes out. Otherwise it goes at the end of the queue like an arriving unit, as shown in figure 6.1.1. The service time is cumulative and there is no loss of time. For example, if a user X needs Y seconds of service where $Y = 2q + e$, and $e < q$, he will get q seconds (the quantum size) in the first pass. The second time, the user gets again q seconds and return to the end of the same queue. On the third pass the user will get only e seconds and will go out of the systems. In this case we will refer to the expected time spent in the queue as $W(t)$, i.e. the expected time spent in the system less the time the unit was being served for a unit requiring t seconds of service.

Parameters and Assumptions. The following assumptions are made .

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad 6.1.01$$

where the mean interarrival time is $1/\lambda$ seconds and the mean arrival rate is λ , units per second.

2. For the service process an exponential service distribution is assumed,

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad 6.1.02$$

where the mean service requirement is $1/\mu$ seconds.

3. The quantum size is a constant value q

4. In a time-sharing system, the user departs from the system as soon as his service requirement is satisfied. Then the amount of time used during the quantum q is distributed according to:

$$F_1(\tau) = \begin{cases} 0, & \tau < 0, \\ 1 - e^{-\mu\tau}, & 0 \leq \tau < q, \\ 1, & \tau \geq q. \end{cases} \quad 6.1.03$$

5. There is an infinite supply of customers.
6. The measure of efficiency is the expected waiting time conditional on the service requirement.
7. There is no time lost for setting up, i.e. an interrupted unit carries with it the knowledge of the processing already done to it.

Analytical Solution.⁹ As it has been said above, our measure of efficiency is the mean waiting time $W(t)$ in the queue as a function of the processing requirement. The procedure to solve the Round-Robin model is to decompose the waiting time in queue into two parts T_1 and T_2 where T_1 is the time necessary to finish the units in service and T_2 is the time necessary to complete the units in the queue at arrival. By finding the expected waiting time in both stages and adding them up one could obtain the expected total waiting time i.e.

$$W_k = E(T_1) + E(T_2)$$

where $E(T_2)$ is the sum of ^Wtime spent in all the queues other than the first one. However, the analysis is made difficult by the fact that the input to the service station is no longer Poisson because of the arrivals joining the queue in the feedback process. We have to resequence the events. The resequencing is justified by the fact that the input process is time-homogeneous and statistically independent of the state of the system.

We have to find the number of units in the queues before the tagged unit, i.e. the length of the queue at the arrival. The queue size distribution is found by altering the model so that the users are served as if there was no limited quantum. Since the limiting queue size distribution

⁹Edward Coffman and Leonard Kleinrock, "Feedback Queuing Model for Time-Shared Systems, Journal of ACM, Vol. 15, No. 4, (October 1968), p. 549-576.

is independent of the queuing discipline, one may obtain the mean queue size ¹⁰. With the queue size one can get $E(T_2)$ i.e. the expected time in the second part of the model.

Adding $E(T_1)$ and $E(T_2)$ one can obtain the mean waiting time dependent on the service required.

The mean waiting time in the Round-Robin model of a unit requiring t seconds of service is: ¹¹

$$W(t) = t + \frac{\rho k q}{1 - \rho} + \frac{(\lambda/2)E_1(\tau^2)}{1 - \beta} [1 - \rho\beta^{k-1}] \quad 6.1.1$$

$$+ \frac{1}{1 - \rho} \left[\frac{\rho^2}{1 - \rho} \frac{1}{\mu} - \frac{\rho q}{1 - \beta} \right] [1 - \beta^k] + \frac{\lambda q e^{-\mu q}}{1 - \beta} \frac{1}{\mu} [1 - \beta^{k-1}],$$

where

$$\rho = \lambda/\mu,$$

$$\beta = \rho + (1 - \rho)e^{-\mu q},$$

where k is the smallest integer such that $kq > t$, where we define $E_k(\tau^2)$ as the second moment of the distribution.

Specifically ,

$$E_1(\tau^2) = \int_0^\infty \tau^2 dF_1(\tau) = \frac{2}{\mu^2} - \frac{e^{-\mu q}}{\mu^2} [\mu^2 q^2 + 2\mu q + 2]. \quad 6.1.2$$

When the quantum size approaches 0 i.e. $q \rightarrow 0$ the model is called a Processor Shared Model. In that model the total time spent in the system for a unit requiring t seconds of service is ¹²

$$W(t) = \frac{t}{1 - \rho} \quad 6.1.3$$

¹⁰ L. Takas, Introduction to The Theory of Queues, Oxford, U. Press, (1961).

¹¹ The proof of this theorem is reproduced in Appendix 10. It has been offered by Edward Coffman and Leonard Kleinrock "Feedback Queuing Models for Time-Shared Systems," Journal of ACM Vol. 15, No. 4, October 1968. pp. 567

¹² Ibid

In the conventional FCFS system the waiting time in the queue is independent of t . The time spent in the system is ¹³

$$W(t) = \frac{\rho(\mu)}{1-\rho} + t = \frac{\lambda}{\mu(\mu-\lambda)} \quad 6.1.4$$

One can see that the unit requiring more than the average amount of service has a longer waiting time in the processor shared system than in the FCFS system.

The mean waiting time in the queue W_k is the mean waiting time in the system $W(t)$ less the service requirement t

$$W_k = W(t) - t \quad 6.1.5$$

The behavior of the model can be shown better with graphical representations. Figure 6.1.3 shows how in the R.R. model the waiting times vary according to the number of quanta. The values used in the example are:

$$\mu = 1.0/\text{sec}$$

$$q = 0.5 \text{ sec}$$

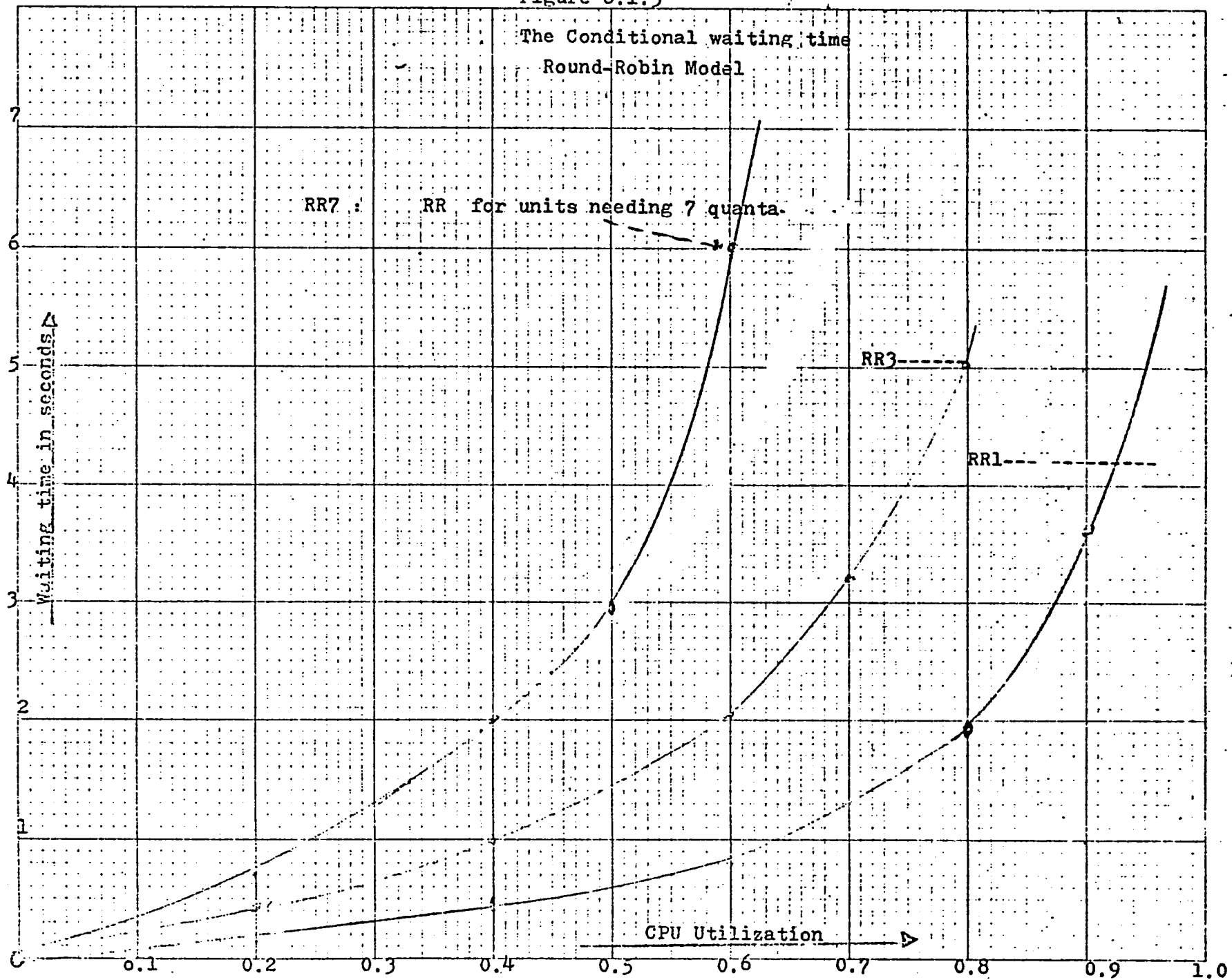
ρ is varied by allowing λ to vary from 0 to 1.0.

The graph has been obtained by replacing the values in equation 6.1.5 and 6.1.2. One can easily see the advantage in that model of units requiring one quantum compare to the units requiring 3 quanta. On the graph, RR 1 indicates the expected waiting time for units requiring one quantum.

The formula implies that the waiting time for ρ approaching 1.0 approaches ∞ . However, units requiring less than one quantum will always be serviced since the units with a required service greater than the quantum will be sent back to the end of the queue. We would like to test the behavior of the model.

¹³ Takacs. Theory of Queues, (Oxford, U Press), 1961, pp.39

Figure 6.1.3



Hypotheses. The following hypotheses will be tested.

- A) The expected waiting time in the Round-Robin model is properly obtained from the analytical solution.
- B) The arrival distribution does not have a significant effect on the expected response time.

Simulation Results. The author tested the above hypotheses using the program TS 50 listed in appendix two. The program generates Poisson arrivals for different values of λ so that ρ varies from 0.2 to 1.0 while μ remains constant at 1.0 unit per second. The quantum is also constant at $q = 0.5$ sec. Preliminary experiments were made by running the model TS 50 listed in appendix two for each level of the CPU utilization so that one could determine the areas where further investigation is needed. Many simulation experiments were made using for the arrival rate, the service time, and the quantum size for the same values as in the analytical results shown in figure 6.1.3. The results have been summarized in figure 6.2.1, figure 6.2.2 and figure 6.2.3. The results of 84 runs are summarized in table 6.2.1 and a sample of the output is shown in tables 6.2.2, 6.2.3, and 6.2.4.

In these examples, in addition to the waiting time and the queue statistics, the tables give for each case a distribution of the time spent in the system. Table XTIME on the results is the distribution of the transit time of all units including the waiting time. Table XTML is the transit time for all units that required more than one quantum. Table XSERV is the % of service over the time in the system. Table PASS is the distribution of the number of passes in the system.

TABLE 6.2.1
WAITING TIME FOR DIFFERENT QUEUE DISCIPLINES

		R1= 0.2	R1= 0.4	R1= 0.6	R1= 0.8	R1= 0.9	R1= 1.0	OVER LOAD
	FCFS	R2= 0.209 WA= .187	R2= 0.425 WA= .572	R2= 0.646 WA= 1.064	R2= 0.861 WA= 2.631	R2= 0.963 WA= 4.015	R2= 1.000 WA=12.508	R2= 1.000 WA=27.667
	FCFS K=1 K=3 K=7	R2= 0.201 W1= .029 W3= .159 W7= .175	R2= 0.406 W1= .165 W3= .509 W7= .334	R2= 0.615 W1= .375 W3= .845 W7= 1.172	R2= 0.827 W1= .897 W3= 1.232 W7= .736	R2= 0.935 W1= 1.586 W3= 1.606 W7= 1.429	R2=1.000 W1= 3.873 W3= 4.683 W7= 4.455	R2= 1.000 W1=16.917 W3=20.783 W7=19.584
	RR	R2= 0.186 WA= .215	R2= 0.374 WA= .462	R2= 0.561 WA= .792	R2= 0.756 WA= 1.835	R2= 0.873 WA= 4.800	R2= 0.941 WA= 9.907	R2= 1.000 WA=21.818
	AV K=1 K=3 K=7	W1= .055 W3= .299 W7= .670	W1= .159 W3= .662 W7= 1.371	W1= .231 W3= 1.073 W7= 1.968	W1= .637 W3= 2.153 W7= 5.651	W1= 1.920 W3= 5.705 W7=14.875	W1= 4.016 W3=12.241 W7=25.690	W1= 7.622 W3=23.446 W7=53.688
	SJF	R2= 0.222 WA= .435	R2= 0.447 WA= .722	R2= 0.674 WA= 1.313	R2= 0.895 WA= 2.004	R2= 0.952 WA= 3.069	R2= 1.000 WA=4.735	R2= 1.000 WA= 6.957
	FB AV K=1 K=3 K=7	R2= 0.206 WA= .328 W1= .159 W3= .288 W7= 2.023	R2= 0.412 WA= .913 W1= .123 W3= .545 W7= 4.139	R2= 0.616 WA= 1.552 W1= .165 W3= 1.309 W7= 4.213	R2= 0.826 WA= 2.157 W1= 0.243 W3= 2.059 W7= 8.800	R2= 0.930 WA= 3.523 W1= .326 W3= 2.152 W7= 8.613	R2= 1.000 WA=11.102 W1= .373 W3= 4.146 W7=80.435	R2= 1.000 WA=19.683 W1= .380 W3=14.830 W7=-----

SYMBOLS USED

R1 = THEORETICAL VALUE OF CPU UTILIZATION

R2 = EXPERIMENTAL VALUE OF CPU UTILIZATION

WA = AVERAGE WAITING TIME

WK = AVERAGE WAITING TIME FOR THE UNITS NEEDING K QUANTA

Q = QUANTUM SIZE = .5 SEC

TIME REQUIRED =1.0 SEC

FCFS = FIRST COME FIRST SERVED

RR = ROUND ROBIN

SJF = SHORTEST JOB FIRST

FB = FEEDBACK OR FOREGROUND BACKGROUND WITH MANY LEVELS OF QUEUES

SOURCE: SUMMARY OF OUTPUT OF THE PROGRAM TS50 LISTED IN APPENDIX TWO

Figure 6.2.1

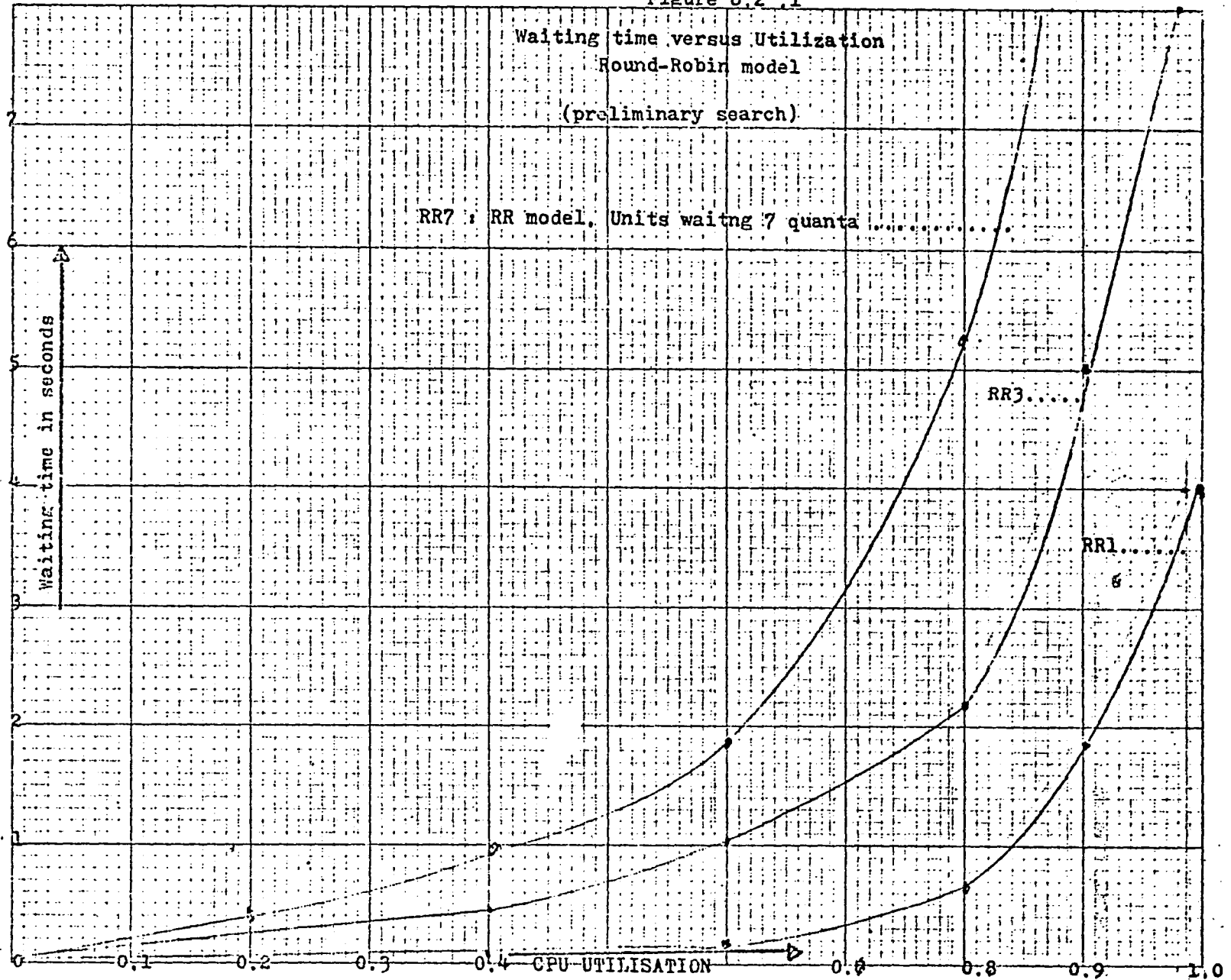


Figure 6.2.2

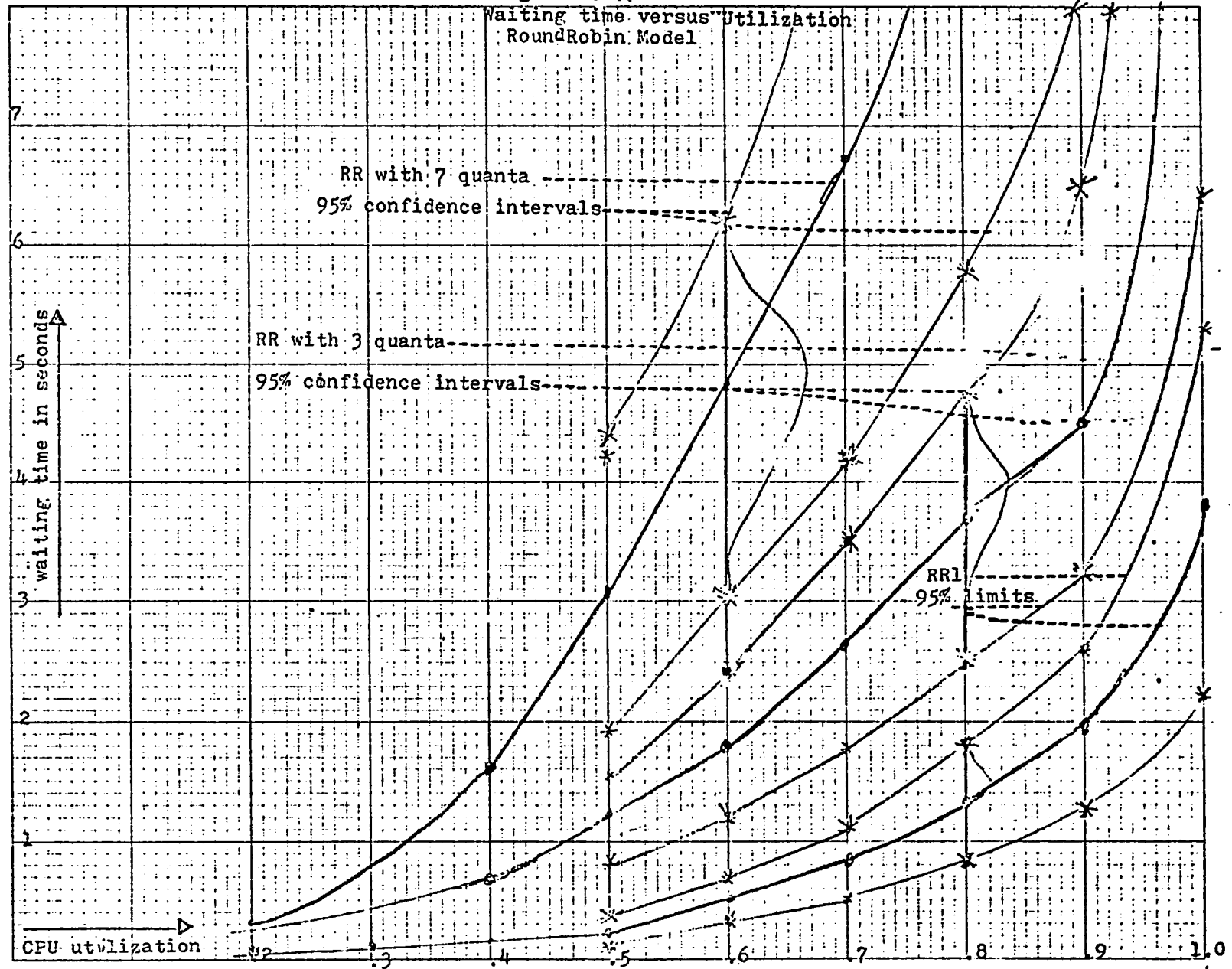
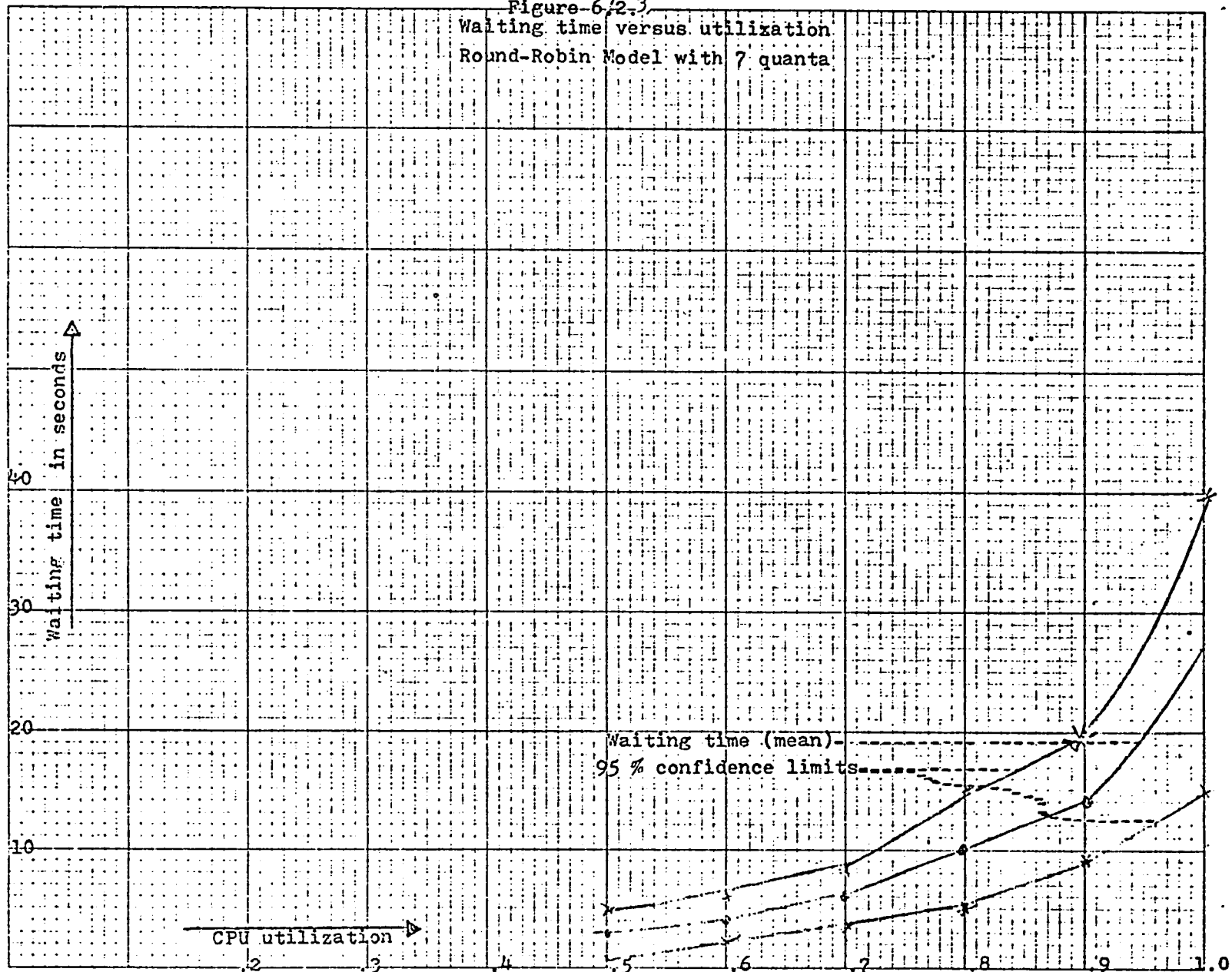


Figure 6.2.3
Waiting time versus utilization
Round-Robin Model with 7 quanta



UTILIZATION OF THE CPU

FACILITY	AVERAGE	NUMBER	AVERAGE	SEIZING	PREEMPTING
CPU	UTILIZATION	ENTRIES	TIME/TRAN	TRANS. NO.	TRANS. NO.
	.922	530	399.594		

WAITING TIME FOR THE CPU

QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
1	3	.297	200	17	8.4	341.074	372.759		
2	7	.770	122	30	24.5	1450.647	1923.684		
3	19	2.709	81	10	12.3	7680.281	8762.011		
4	17	2.200	54	5	9.2	9356.367	10311.191		
5	20	3.485	32	3	9.3	25003.968	27590.585		
6	14	2.194	17	1	5.8	29639.234	31491.687		
7	10	.509	12		.0	9749.914	9749.914		
8	5	.077	5		.0	3545.399	3545.399		
9	3	.078	4		.0	4519.750	4519.750		
10	1	.001	1		.0	231.000	231.000		
11	1	.000	1	1	100.0	.000	.000		
12	1	.000	1	1	100.0	.000	.000		

AVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

TIME SPENT IN THE SYSTEM for all units using more than one quantum

TABLE	XTM1	MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS	NON-WEIGHTED	
ENTRIES IN TABLE	122	24582.187	37680.000	2999027.000		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-.652
500	0	.00	.0	100.0	.020	-.639
1000	6	4.91	4.9	95.2	.040	-.625
1500	11	9.01	13.9	86.0	.061	-.612
2000	7	5.73	19.6	80.3	.081	-.599
2500	8	6.55	26.2	73.7	.101	-.586
3000	6	4.91	31.1	68.8	.122	-.572
3500	6	4.91	36.0	63.9	.142	-.559
4000	6	4.91	40.9	59.0	.162	-.546
4500	3	2.45	43.4	56.5	.183	-.532
5000	2	1.63	45.0	54.9	.203	-.519
5500	1	.81	45.9	54.0	.223	-.506
6000	4	3.27	49.1	50.8	.244	-.493
6500	2	1.63	50.8	49.1	.264	-.479
7000	2	1.63	52.4	47.5	.284	-.466
7500	4	3.27	55.7	44.2	.305	-.453
8000	2	1.63	57.3	42.6	.325	-.440
8500	0	.00	57.3	42.6	.345	-.426
9000	0	.00	57.3	42.6	.366	-.413
9500	0	.00	57.3	42.6	.386	-.400
10000	3	2.45	59.8	40.1	.406	-.387
10500	1	.81	60.6	39.3	.427	-.373
11000	1	.81	61.4	38.5	.447	-.360
11500	0	.00	61.4	38.5	.467	-.347
12000	2	1.63	63.1	36.8	.488	-.333
12500	1	.81	63.9	36.0	.508	-.320
13000	0	.00	63.9	36.0	.528	-.307
13500	0	.00	63.9	36.0	.549	-.294
14000	3	2.45	66.3	33.6	.569	-.280
OVERFLOW	41	33.60	100.0	.0		

Note: This is a sample only many similar runs were made.

Table: 6.2.2

TABLE XTIME
ENTRIES IN TABLE
200

Time in the system for all units.

MEAN ARGUMENT
15206.777

STANDARD DEVIATION
31648.000

SUM OF ARGUMENTS
3041356.000

NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-.480
200	14	6.99	6.9	93.0	.013	-.474
400	18	8.99	15.9	84.0	.026	-.467
600	17	8.49	24.4	75.5	.039	-.461
800	17	8.49	32.9	67.0	.052	-.455
1000	7	3.49	36.4	63.5	.065	-.448
1200	11	5.49	41.9	58.0	.078	-.442
1400	8	3.99	45.9	54.0	.092	-.436
1600	5	2.49	48.4	51.5	.105	-.429
1800	1	.49	48.9	51.0	.118	-.423
2000	4	1.99	50.9	49.0	.131	-.417
2200	2	.99	51.9	48.0	.144	-.410
2400	5	2.49	54.4	45.5	.157	-.404
2600	1	.49	54.9	45.0	.170	-.398
2800	3	1.49	56.4	43.5	.184	-.392
3000	3	1.49	57.9	42.0	.197	-.385
3200	3	1.49	59.4	40.5	.210	-.379
3400	0	.00	59.4	40.5	.223	-.373
3600	4	1.99	61.4	38.5	.236	-.366
3800	4	1.99	63.4	36.5	.249	-.360
4000	1	.49	63.9	36.0	.263	-.354
4200	1	.49	64.4	35.5	.276	-.347
4400	1	.49	64.9	35.0	.289	-.341
4600	1	.49	65.4	34.5	.302	-.335
4800	0	.00	65.4	34.5	.315	-.328
5000	2	.99	66.4	33.5	.328	-.322
5200	1	.49	66.9	33.0	.341	-.316
5400	0	.00	66.9	33.0	.355	-.309
5600	1	.49	67.4	32.5	.368	-.303
OVERFLOW	65	32.49	100.0	.0		
AVERAGE VALUE OF OVERFLOW		43982.89				

TABLE PASS
ENTRIES IN TABLE
200

number of passes required.

MEAN ARGUMENT
2.649

STANDARD DEVIATION
1.980

SUM OF ARGUMENTS
530.000

NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-1.338
1	78	38.99	38.9	61.0	.377	-.833
2	41	20.49	59.4	40.5	.754	-.328
3	27	13.49	72.9	27.0	1.132	.176
4	22	10.99	83.9	16.0	1.509	.681
5	15	7.49	91.4	8.5	1.886	1.186
6	5	2.49	93.9	6.0	2.264	1.691
7	7	3.49	97.4	2.5	2.641	2.196
8	1	.49	97.9	2.0	3.018	2.701
9	3	1.49	99.4	.5	3.396	3.206
10	0	.00	99.4	.5	3.773	3.711
11	0	.00	99.4	.5	4.150	4.216
12	1	.49	100.0	.0	4.528	4.721

REMAINING FREQUENCIES ARE ALL ZERO

Note: this is only a sample out of many similar runs.

TABLE 6.2.3

Source: Output of the program TS50 written by the author and listed in Appendix two.

111

% of service time out of the time in the system.

TABLE XSER ENTRIES IN TABLE 200		MEAN ARGUMENT 35.375	STANDARD DEVIATION 29.530	SUM OF ARGUMENTS 7075.000		NON-WEIGHTED
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-1.199
10	49	24.49	24.4	75.5	.282	-.860
20	34	16.99	41.4	58.5	.565	-.521
30	23	11.49	52.9	47.0	.848	-.182
40	17	8.49	61.4	38.5	1.130	.156
50	22	10.99	72.4	27.5	1.413	.495
60	17	8.49	80.9	19.0	1.696	.834
70	9	4.49	85.4	14.5	1.978	1.173
80	7	3.49	88.9	11.0	2.261	1.512
90	4	1.99	90.9	9.0	2.544	1.851
100	18	8.99	100.0	.0	2.826	2.190

REMAINING FREQUENCIES ARE ALL ZERO

TOTAL WAITING TIME IN QUEUE

CONTENTS OF FULLWORD SAVEVALUES (NON-ZERO)

SAVEVALUE	NR.	VALUE	NR.	VALUE	NR.	VALUE	NR.	VALUE	NR.	VALUE
1		341	2	1791	3	9471	4	18827	5	43830

Note: This is only a sample out of many similar runs.

Table 6.2.4

Source: Output of the program TS50 written by the author and listed in Appendix two.

The preliminary results are shown in graph form on figure 6.2.1 using the data given on table 6.2.1. The reader should recall that the preliminary runs are made in order to get an indication of the areas that should be investigated further i.e. the areas where we should get sufficient data for and where we can build confidence intervals. In the preliminary search many simulation runs were made using different arrival rates.

It was then found that the areas of high CPU utilizations should be investigated further by building confidence intervals. For each utilization value, 12 independent runs of 200 non-transient observations were made so that we can build confidence limits. The results are summarized in table 6.2.5. The table shows the mean waiting time in queue for units requiring 7, 3, and 1 quanta of service. In order to make inference about the population, 95% and 99% confidence intervals are shown in tables 6.2.6, 6.2.7, 6.2.8, and 6.2.9.

The behavior of the system is illustrated in figures 6.2.2, 6.2.3, and 6.2.4. In the figures, RR_n indicates the expected waiting time for units requiring exactly n quanta of service. We have limited ourselves to illustrate as we have done in the analytical solution in figure 6.1.3 only the three different groups of requests i.e. RR_1 , RR_3 , RR_7 for units requiring 1, 3, and 7 quanta. Since we used a constant quantum of 0.5 second, if a unit requires 3 quanta it means that it required between 1.0001 and 1.5 second. Figure 6.2.2 shows the mean conditional waiting time for units requiring 1, 3, and 7 quanta. Comparing figures 6.2.2, 6.1.3, and 6.2.4 one can see that for a utilization factor of less than 0.8 the waiting time obtained from analytical techniques for the RR_1 model are within the confidence limits of the simulation results. However, for $\rho > 0.8$ the expected waiting

SUMMARY OF WAITING TIME IN QUEUE ROUND ROBIN MODEL - ANY NUMBER OF LEVELS												
CPU UTIL.	RR1				RR3				RR7			
1.0	5363	01131	2771	1956	16265	3485	8525	6255	38379	8413	17181	18627
	5789	3278	3226	1182	17452	10551	9925	3888	44616	25370	23846	09209
	3923	2957	3867	10251	11577	8928	11905	31395	28752	18167	26380	75991
0.9	1691	3524	906	1912	5053	10894	2862	6347	12324	25832	6656	13515
	1567	4258	1335	1560	5011	12636	4379	5298	14948	35529	11301	13726
	764	2357	1564	2357	2403	7265	5032	7265	5238	99016	10297	18801
0.8	838	2163	710	1479	2665	6873	2214	4907	6752	15937	5312	15094
	1050	2540	563	1064	3864	7572	1858	3766	11429	20938	4585	9614
	543	1685	644	0920	1784	5155	2047	3123	3696	12755	4467	6107
0.7	459	1513	578	1185	1681	4972	1766	4065	4285	11359	4250	11224
	750	1554	377	823	2600	4932	1212	2917	7461	14081	3096	8050
	401	1027	398	544	1363	3334	1287	2259	2738	7594	3087	3934
0.6	261	1071	442	881	1044	3606	1396	3312	2789	8420	3696	9606
	490	742	239	612	1622	2515	816	2194	5536	6637	2214	5894
	286	606	266	394	1083	2166	17	1579	2361	4644	2219	2635
0.5	174	627	327	572	754	2220	1122	2374	1877	5732	2815	7475
	339	325	130	360	1222	1300	567	1344	4473	3388	1288	1822
	228	298	195	270	924	1260	583	1166	2366	2889	1767	1703

TABLE: 6.2.5

SOURCE: OUTPUT OF 72 RUNS OF THE PROGRAM TS50 LISTED IN APPENDIX TWO.

CONFIDENCE LIMITS FOR WAITING TIME MODEL RR1 UNITS REQUIRING 1 QUANTUM				
CPU R	MEAN OF 12 RUNS	INTERVALS		
		95%	99%	
1.0	3808	2232 - 5383	1590 - 6025	
0.9	1985	1332 - 2639	1066 - 2704	
0.8	1349	833 - 1866	623 - 2176	
0.7	800	529 - 1071	418 - 1182	
0.6	524	354 - 694	285 - 763	
0.5	320	225 - 414	187 - 453	

TABLE: 6.2.6.

CONFIDENCE LIMITS FOR WAITING TIME MODEL RR3 UNITS REQUIRING 3 QUANTA				
CPU R	MEAN OF 12 RUNS	INTERVALS		
		95%	99%	
1.0	11429	6440 - 16418	4408 - 18450	
0.9	4531	3265 - 5797	2749 - 6613	
0.8	3820	2581 - 5059	2077 - 5564	
0.7	2690	1832 - 3549	1462 - 3998	
0.6	1845	1253 - 2438	1011 - 2679	

TABLE: 6.2.7.

CONFIDENCE LIMITS FOR WAITING TIME MODEL RR7 UNITS REQUIRING 7 QUANTA				
CPU R	MEAN OF 12 RUNS	INTERVALS		
		95%	99%	
0.5	1236	878 - 1594	732 - 1740	
1.0	2791	16188 - 39633	11413 - 44408	
0.9	14727	9347 - 20107	7156 - 22298	
0.8	10473	5735 - 15212	3805 - 17147	
0.7	6761	4332 - 9190	3343 - 10180	
0.6	4720	3113 - 6327	2459 - 6982	
0.5	3132	1740 - 4325	1454 - 4811	

TABLE: 6.2.8

SOURCE: OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR
AND LISTED IN APPENDIX SEVEN.

time obtained from simulation is significantly less than the one obtained from analytical techniques. We fail to accept the hypothesis. In order to show what happen to the RR7 as ρ approaches 1.0 we have shown in a different scale graph the mean waiting time for high utilization values. The expected waiting time is large but less than infinity. The reader will recall that there are in the system some units requiring more than 7 quanta. Since the units requiring more than 7 quanta are interrupted and sent back to the queue, units requiring less quanta have a chance to be serviced.

One should recall that in the analytical solution, many assumptions were made. The total waiting time was broken into two parts. The expected waiting time for each part was found and the results added up. The analytical solution also implied

ROUND-ROBIN			
ONE-SIDED CONFIDENCE LIMITS UNITS REQUIRING 1 QUANTUM			
CPU R	MEAN OF 12 RUNS	LIMITS	
		95%	99%
1.0	3808	5093	5740
0.9	1985	2518	2614
0.8	1349	1770	2072
0.7	800	1021	1134
0.6	524	662	733
0.5	320	395	435

TABLE: 6.2.9

Figure 6.2.2

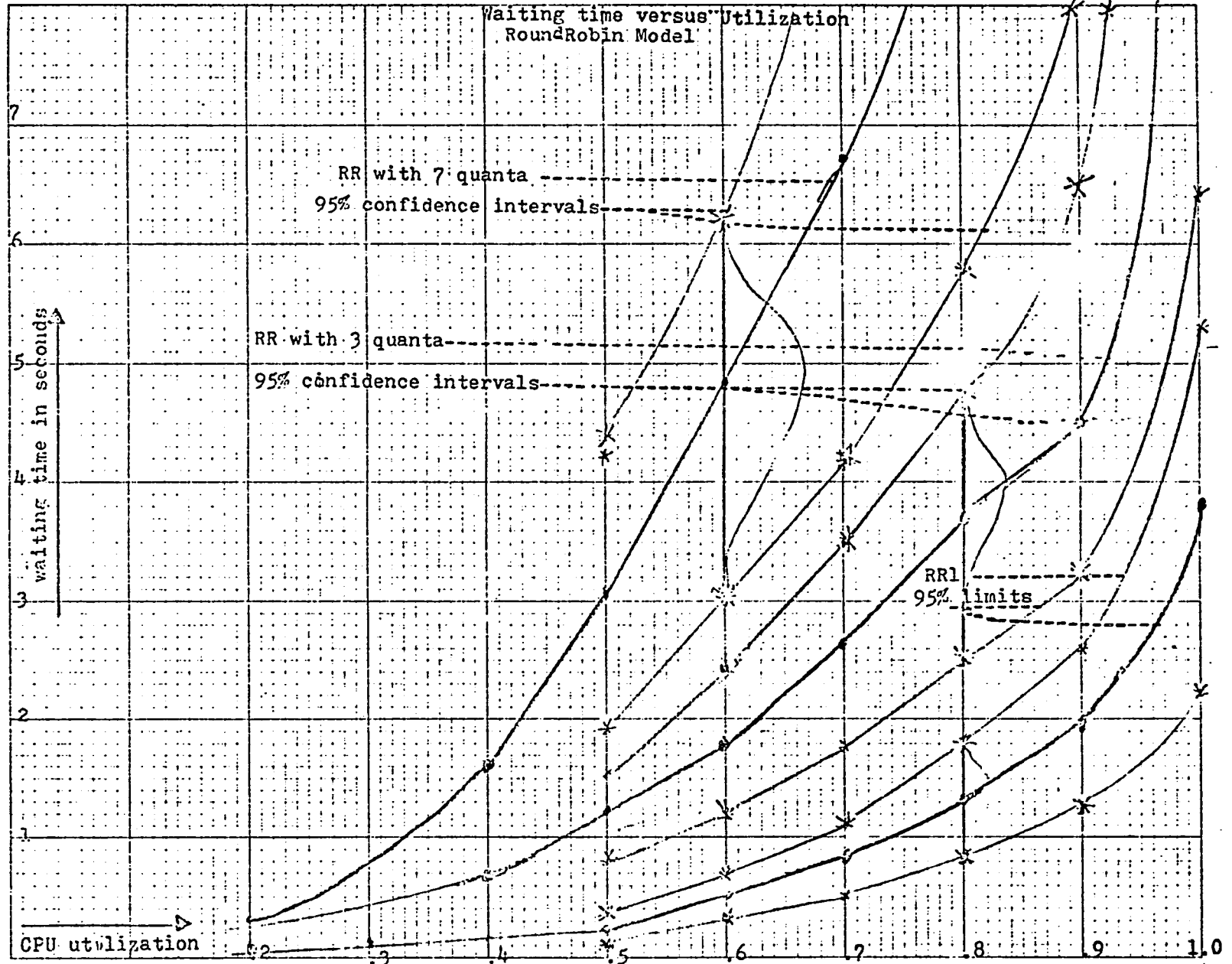


Figure 6.2.3
Waiting time versus utilization
Round-Robin Model with 7 quanta

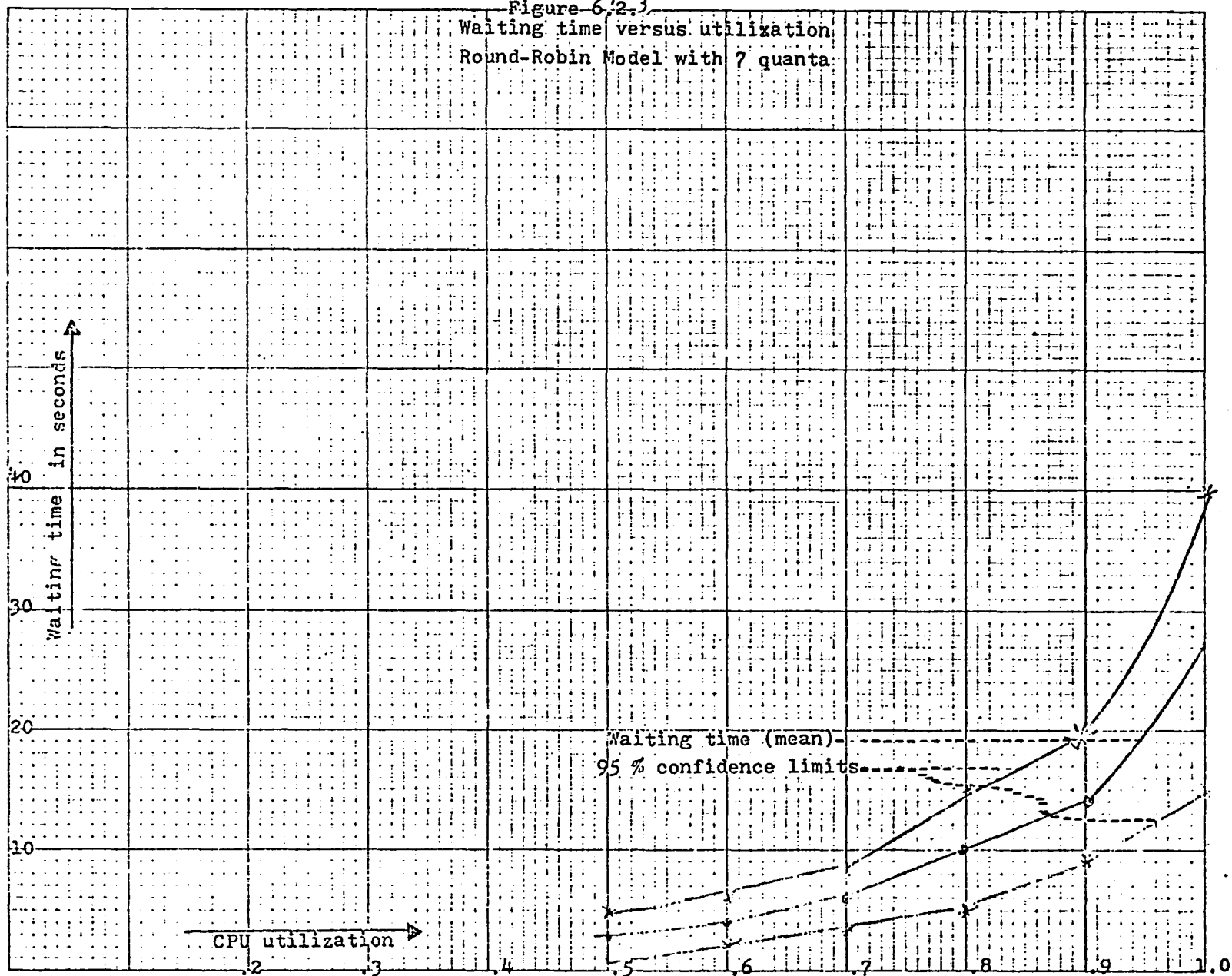
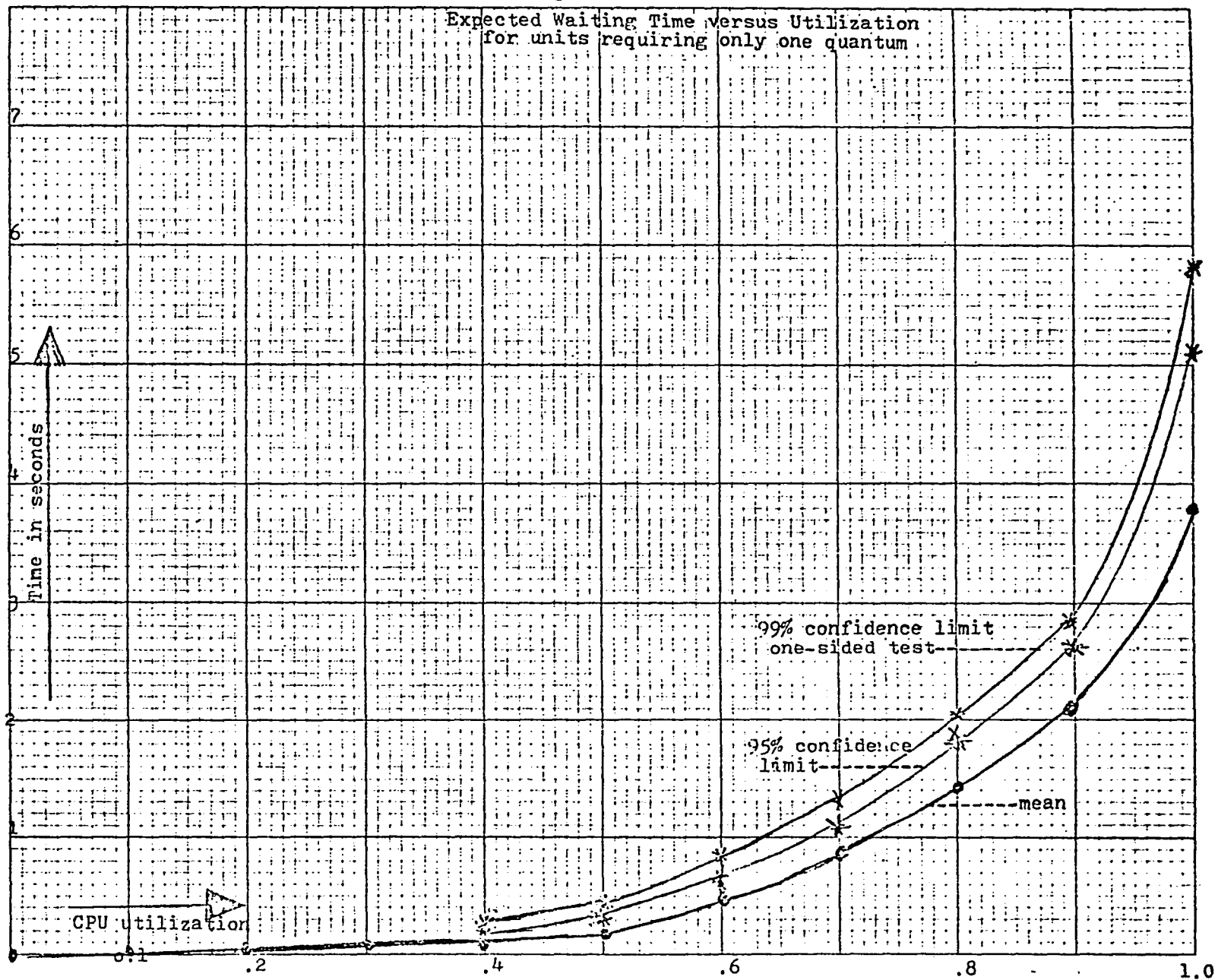


Figure 6.2.4



the assumption that the resequencing of events has no effect on the waiting time provided we take into consideration all events. An assumption was especially made regarding the order in which the events take place at the end of a time interval. The analytical solution also assumed the existence of two systems. First, in the "late-arrival" system, a unit is ejected from service and sent back to the end of the queue just after a unit has arrived. Second, in the "early-arrival" system the events occur in a reversed manner. It was also assumed that all units will join the queue at the beginning of a time interval. Finally, a critical assumption was made concerning the arrival distribution in the queue. One should recall that two types of units will join the queue, the regular arrivals and the units which have been interrupted at the end of a quantum.

Testing The Second Hypothesis. The author ran simulation experiments for^{7th} Round-Robin model with far apart arrival distributions such as Poisson, Normal, and Uniform. Because of the expenses involved, the results were obtained only for high CPU utilization as shown on tables 6.2.10 and 6.2.11. The first table gives the expected response time for all units while the second table gives the expected response time for units requiring more than one quantum.

An analysis of variance was performed on the data. The results are given in tables 6.2.12 and 6.2.13. It can easily be seen that the arrival distribution does have an effect on the expected response time in both cases, that is for units requiring more than one quantum as well as for all units. However, it can be seen that the inter-action of the utilization factor and the arrival pattern has no significant effect.

EXPECTED RESPONSE TIME
FOR ALL UNITS

CPU	POISSON			NORMAL			UNIFORM		
1.0	18496	9593	3887	12344	9359	3100	8882	14100	4376
	05746	5870	11987	4865	5296	8926	6209	6797	10226
	10663	7028	5928	5764	9789	4126	3038	11577	4696
	8744	7162	9223	8075	6050	4497	8672	7024	9248
0.9	5556	09461	2688	4113	6491	2014	3835	09222	3482
	4813	6503	2673	3464	6163	2078	2233	8612	2744
	7326	4436	5844	4771	3064	3027	3843	4008	6282
0.8	5276	06965	2256	2406	3495	1816	2666	4971	2747
	3701	4237	7225	2792	2442	3513	3987	2353	3382
	2535	4798	2157	1691	4460	1319	1713	5724	1933
	5361	2648	3999	2795	2095	2524	2625	2181	3012
0.7	2582	5253	7923	1819	2400	1576	2005	3033	2105
	3359	3239	7389	2133	1607	2131	2718	1831	2466
	3885	3207	3861	2109	2013	1809	1982	1706	2436

TABLE: 6.2.10

SOURCE: SUMMARY OF 144 RUNS OF THE PROGRAM TS50 LISTED IN
APPENDIX TWO.

NOTE: ALL TIMES ARE IN MILLISECONDS.

EXPECTED RESPONSE TIME
FOR UNITS REQUIRING MANY QUANTA

CPU	POISSON			NORMAL			UNIFORM		
1.0	36605	16837	06863	17799	13893	4318	12604	20351	6173
	11239	9273	18980	6989	7062	11888	8919	08989	13658
	16198	11505	10004	3860	14178	6156	4670	16890	6896
	14887	12188	14945	11010	9100	6003	12742	6704	12028
.9	8997	13908	3724	6018	9562	3160	5577	13832	4908
	6612	7107	13596	5831	4013	8415	7004	5511	9315
	7064	9869	3956	3786	9093	3267	3494	12719	4088
	10187	6765	7724	6446	4755	4114	5682	4128	9027
.8	3503	10165	3106	3653	5184	2599	3999	7118	3886
	5333	5857	9472	4129	3436	4702	5796	3274	4486
	3826	7361	3257	2628	6716	2075	2695	9502	2970
	7785	4141	5329	3982	2804	3524	4083	2982	4284
0.7	3976	7716	2614	2852	3600	2262	3116	4469	2973
	4948	4550	5532	3159	2253	2865	3954	2564	3324
	3016	3908	1854	2209	4975	1831	2223	6370	2498
	5444	2055	2847	3202	3001	2363	3052	3500	3278

TABLE: 6.2.11

SOURCE: SUMMARY OF 144 RUNS OF THE PROGRAM TS50 LISTED IN
APPENDIX TWO.

NOTE: ALL TIMES ARE IN MILLISECONDS.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME RR MODEL

ANOVA--LEAST SQUARES SOLUTION				
SOURCE	SS	DF	MS	F
A	1446522880.000	3.	482174208.000	19.021
S(A)	1115381760.000	44.	25349584.000	
B	198529024.000	2.	99264512.000	19.162
AB	76093200.000	6.	12680533.000	2.448
BS(A)	455868416.000	88.	5180322.000	
GRAND MEAN = 6719.502				
ANOV--UNWEIGHTED SOLUTION				
SOURCE	SS	DF	MS	F
A	1446546688.000	3.	482192144.000	19.021
S(A)	1115381760.000	44.	25349584.000	
B	198552608.000	2.	99276304.000	19.164
AB	76065808.000	6.	12677634.000	2.447
BS(A)	455868416.000	88.	5180322.000	
GRAND MEAN = 6719.570				
CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B				
	1	2	3	
1	14959.500	9353.832	10876.082	
2	8292.414	5712.500	7107.082	
3	5755.250	3785.750	4504.582	
4	4038.250	2806.333	3443.417	

NOTE: ONLY THE UNITS REQUIRING MORE THAN ONE QUANTUM HAVE BEEN INCLUDED.

A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION ; BOTH ARE FIXED.

TABLE: 6.2.12

SOURCE: OUTPUT OF THE PROGRAM FINADV LISTED IN APPENDIX FIVE.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME RR MODEL

ANOVA--LEAST SQUARES SOLUTION				
SOURCE	SS	DF	MS	F
A	572396544.000	3.	190798848.000	19.319
S(A)	434553088.000	44.	9876206.000	
B	82515968.000	2.	41257984.000	23.368
AB	3272960.000	6.	545493.313	0.309
BS(A)	155368704.000	88.	1765553.000	
GRAND MEAN = 4715.414				
ANOVA--UNWEIGHTED SOLUTION				
SOURCE	SS	DF	MS	F
A	572395520.000	3.	190798496.000	19.319
S(A)	434553088.000	44.	9876206.000	
B	82512976.000	2.	41256480.000	23.367
AB	3266496.000	6.	544416.000	0.308
BS(A)	155368704.000	88.	1765553.000	
GRAND MEAN = 4715.414				
CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B				
	1	2	3	
1	8777.250	6849.250	7969.914	
2	5771.332	4099.914	4938.250	
3	4481.750	2580.583	3079.167	
4	3797.833	1951.250	2233.500	

NOTE: ALL UNITS HAVE BEEN INCLUDED WITHOUT REFERENCE TO SERVICE NEEDED.
 A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION ; BOTH ARE FIXED.

TABLE: 6.2.13

SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

THE FEEDBACK MODEL

The Model. In the model the arriving unit joins a queue from which it will be served. The unit is allocated a limited quantum q as in the Round Robin. However, if the service is not completed during the quantum, the units go to the end of a second queue with lower priority as shown in figure 6.3.1. This second queue is empty. If the service is not completed after the second pass i.e. after coming from the queue of second priority, the unit goes to the end of a third queue of even lower priority. The model where there is no limit on the number of levels will be referred to as FB_∞ or FB. Sometimes there is a limit on the maximum number of queues. In that case we will refer to it as FB(n). A two FB(2) is often referred^u as a Foreground Background model. If n is finite, the n^{th} queue is served usually like Round Robin. However, some models are such that the service is completed in the n^{th} queue, i.e. the quantum size is then changed.

Parameters and Assumptions. The following assumptions are made :

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad 6.1.01$$

where the mean interarrival time is $1/\lambda$ seconds and the mean arrival rate is λ , units per second.

2. For the service process an exponential service distribution is assumed

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad 6.1.02$$

where the mean service requirement is $1/\mu$ seconds.

3. The quantum size is a constant value q

4. In a time-sharing system, the user departs from the system as soon as his service requirement is satisfied. Then the amount of time used during the quantum q is distributed according to:

$$F_1(\tau) = \begin{cases} 0, & \tau < 0, \\ 1 - e^{-\mu\tau}, & 0 \leq \tau < q, \\ 1, & \tau \geq q. \end{cases} \quad 6.1.03$$

5. There is an infinite supply of customers.
6. The measure of efficiency is the expected waiting time conditional on the service requirement.
7. There is no time lost for setting up, i.e. an interrupted unit carries with it the knowledge of the processing already done to it.
8. There is an infinite number of levels, i.e. there are enough queues so that they all have a priority less than the preceding one or n is large enough so that we need only $n - 1$ queues.

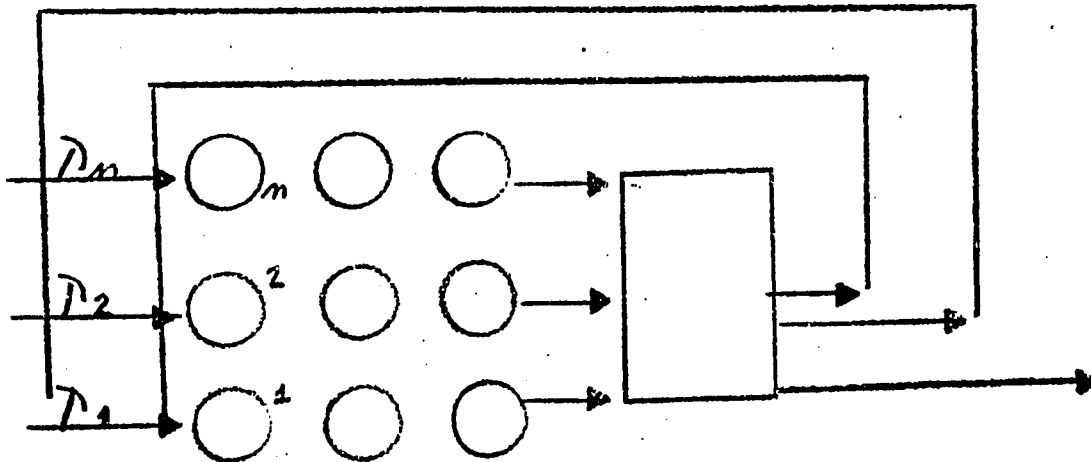
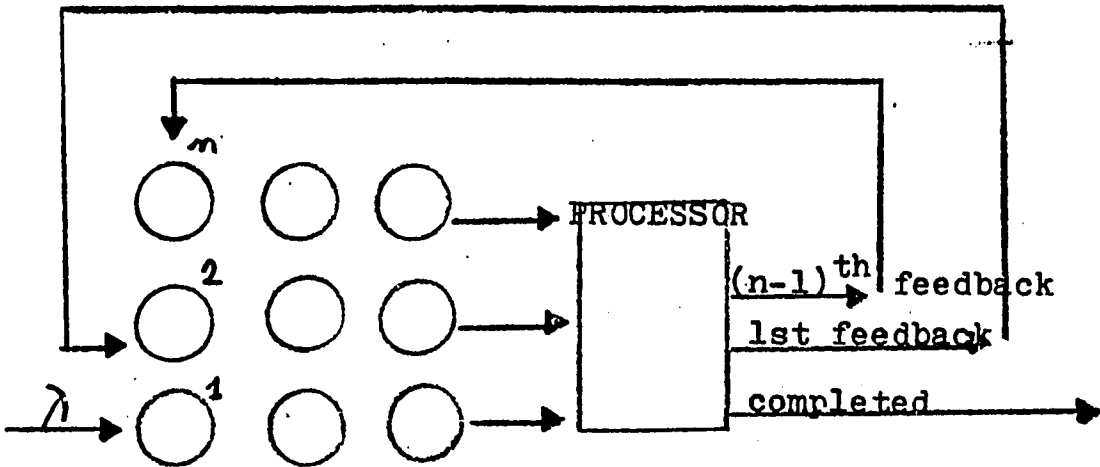


Figure 6.3.1 - The model with n levels of Queues

(upper without priority; lower case with priority.)

Source: Modification of the figures presented in:

E. Coffman and L. Kleinrock, "Feedback Queuing Model for Time Shared Systems", Journal of ACM, Vol. 15, No. 4. (October 1969), p. 554.

Analytical Solution. The measure of the system is again the expected waiting time conditional on the amount of service required. As indicated above, a unit ready for service in any queue level will not be serviced unless all lower level queues are empty. A result for $FB(n)$ has been obtained by Coffman and Kleinrock¹⁴. The $FB(n)$ model is modified such that in the n^{th} pass instead of being a Round Robin, the complete service is given to the user i.e. the quantum is increased. However, if n is large or larger than needed i.e. n approaches ∞ , this restriction will not make any difference since there will never be any unit in the n^{th} queue. The total waiting time in queue is broken down into two parts when a unit arrives in a system. In equilibrium we may consider T_1 the time to complete the unit in service plus the time to complete the units in the first k queues where $kq > t$, i.e. the arriving unit needs more than k queues. We may also consider T_2 the time to process new arrivals that occurs while our unit is moving to lower queues. One should remember that the new unit will be served in queue 1 for a quantum size before any units already in other queues. The two levels are solved and the two expected waiting times are added up.

A simplification is needed. As we said above, the arrival to the high priority queue i.e. the first queue is Poisson. However, the arrival to the lower priority or to other queues is not Poisson as pointed out by Schrage¹⁵,

¹⁴ Kleinrock and Coffman. op. cit. pp. 558

¹⁵ L. E. Schrage, "The Queue M/G/1 with Feedback to Lower Priority Queues", Management Science, Vol. 13, No. 7, (March 1967), pp. 466-474

The solution expressed by Kleinrock assumes that the arrivals at lower queues are Poisson claiming that for the average waiting time it is unimportant, as expressed by Miller and Schrage.¹⁶ Having obtained a two level model and having made the necessary assumption the FB(n) model can be solved in terms of the expected waiting time. We first find the expected total time in the system $W(t)$ and from there obtain the expected total time in queue W_k . The proof has been offered by Kleinrock and Coffman¹⁷ and it is reproduced in appendix seven. The following results have been obtained.

$$W(t) = \frac{(\lambda/2)[E_k(\tau^2) + \gamma_k E_1(\tau^2)]}{[1 - \rho(1 - e^{-\mu k q})][1 - \rho(1 - e^{-\mu(k-1)q})]} + \frac{\rho(1 - e^{-\mu(k-1)q})}{1 - \rho(1 - e^{-\mu(k-1)q})} (k-1)q + t, \quad 1 \leq k \leq N-1, \quad 6.3.1$$

$$W(t) = \frac{\rho(1/\mu)}{(1 - \rho)[1 - \rho(1 - e^{-\mu(N-1)q})]} + \frac{\rho(1 - e^{-\mu(N-1)q})}{1 - \rho(1 - e^{-\mu(N-1)q})} (k-1)q + t, \quad k \geq N,$$

where k is the smallest integer such that $kq > t$, where we define $E_k(\tau^2)$ as the second moment of the distribution

$$F_k(\tau) = \begin{cases} 0, & \tau < 0, \\ 1 - e^{-\mu\tau}, & 0 \leq \tau < kq, \\ 1, & \tau \geq kq, \end{cases}$$

with

$$E_k(\tau) = \frac{1}{\mu} [1 - e^{-\mu k q}],$$

$$E_k(\tau^2) = \frac{2}{\mu^2} - \frac{e^{-\mu k q}}{\mu^2} [(\mu k q)^2 + 2\mu k q + 2],$$

and where

$$\gamma_k = \frac{e^{-\mu k q}}{1 - e^{-\mu q}}.$$

¹⁶ L. W. Miller and L. E. Schrage, "The Queue M/G/1 with shortest Remaining Processing Time Discipline," Operations Research, Vol. 14, No. 9 (July 1966)

¹⁷ The proof is given in Appendix Eleven. It has been taken from Kleinrock and Coffman, op.cit. pp. 558-559

The waiting time in queue is

$$W_k = w(t) - t$$

6.3.2

where k is the number of quanta or the number of passes through the system

$$(k-1)q < t \leq kq$$

The behavior of the system can be shown better by an example. Using the parameters as in the Round Robin model we have:

$$\mu = 1.0/\text{sec}$$

$$q = 0.5/\text{sec}$$

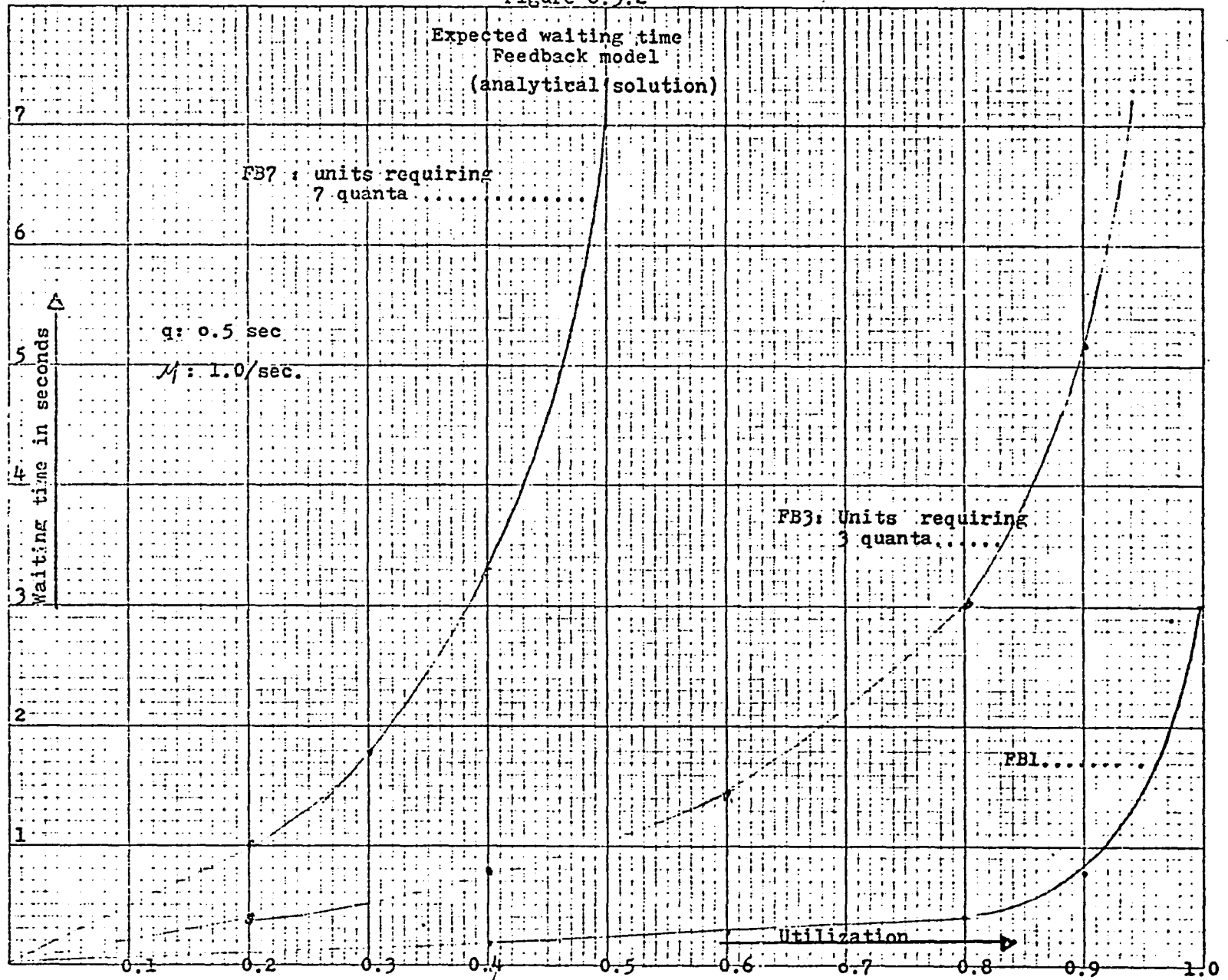
$$n = \infty$$

$$\rho \text{ varying from } 2 \text{ to } 1.0$$

Figure 6.3.2 obtained by replacing the values in the equation 6.3.1 and 6.3.2 shows the mean waiting time for units requiring 1, 3, and 7 quanta. The units with shorter requirements enjoy shorter waiting time than the other units. As one recalls from chapter two or can see on figures 6.1.1 and 6.3.1, in the FB model no unit of shorter service time than q will have to wait behind a unit with service time longer than q . In the case of multiple level of FB, there is no unit of shorter service waiting behind a longer service time unit. Thus we shall discuss later in this chapter the relationship between the behavior of the RR and FB models.

The reader will recall that the analytical solution is based on the assumption that the arrival distribution on lower queues has no effect on the mean waiting time. We are interested in testing the results so obtained and we will make an inquiry into the arrival distribution at the high level queue, i.e. the arrival distribution in the system. We will then test the following hypotheses.

Figure 6.3.2



- Hypotheses.
- a. The expected total waiting time is properly obtained from the analytical solution for the FB model.
 - b. The arrival distribution does not have a significant effect on the expected total response time.

Simulation Results. The author tested the above hypotheses using the program TS50 listed in appendix two and described above. The parameters used are the same as the one used in the Round Robin model. As explained above preliminary experiments were made to determine the areas where research should be conducted. In the preliminary runs the author did not take confidence intervals. One run of 200 non-transient arriving units was made. Doing a preliminary search as described in the previous section we obtained the results shown in table 6.4.1 which is the same as 6.2.1. The mean waiting time for the FB model ^{are} graphed in figure 6.4.1. Comparing the graph 6.3.2 and 6.4.1 one can see that the behavior of the FB₁ units i.e. of the user who need less than one quantum is different when ρ is close to 1.0. One can also see that there are plateaux in FB₇ and FB₃ for $4 < \rho < 5$ and $8 < \rho < 9$ respectively. One can also see that for

$\rho > 4$ the average waiting time seems to differ from the analytical results. This preliminary search indicates that we should investigate further i.e. build confidence intervals with more experiments. We should also investigate points in-between where a plateau seems to occur.

TABLE 6.4.1
WAITING TIME FOR DIFFERENT QUEUE DISCIPLINES

		R1= 0.2	R1= 0.4	R1= 0.6	R1= 0.8	R1= 0.9	R1= 1.0	OVER LOAD
	FCFS	R2= 0.209 WA= .187	R2= 0.425 WA= .572	R2= 0.646 WA= 1.054	R2= 0.861 WA= 2.631	R2= 0.963 WA= 4.015	R2= 1.000 WA=12.508	R2= 1.000 WA=27.667
	FCFS K=1 K=3 K=7	R2= 0.201 W1= .029 W3= .159 W7= .175	R2= 0.406 W1= .165 W3= .509 W7= .334	R2= 0.615 W1= .375 W3= .845 W7= 1.172	R2= 0.827 W1= .897 W3= 1.232 W7= .736	R2= 0.935 W1= 1.586 W3= 1.606 W7= 1.429	R2=1.000 W1= 3.873 W3= 4.683 W7= 4.455	R2= 1.000 W1=16.917 W3=20.793 W7=19.584
	RR	R2= 0.136 WA= .215	R2= 0.374 WA= .462	R2= 0.561 WA= .792	R2= 0.756 WA= 1.835	R2= 0.873 WA= 4.800	R2= 0.941 WA= 9.907	R2= 1.000 WA=21.818
	AV K=1 K=3 K=7	W1= .055 W3= .299 W7= .670	W1= .159 W3= .662 W7= 1.371	W1= .231 W3= 1.073 W7= 1.968	W1= .637 W3= 2.153 W7= 5.651	W1= 1.920 W3= 5.705 W7=14.875	W1= 4.016 W3=12.241 W7=25.690	W1= 7.622 W3=23.446 W7=53.693
	SJF	R2= 0.222 WA= .435	R2= 0.447 WA= .722	R2= 0.674 WA= 1.313	R2= 0.895 WA= 2.004	R2= 0.952 WA= 3.069	R2= 1.000 WA=4.735	R2= 1.000 WA= 6.957
	FB	R2= 0.206 WA= .328	R2= 0.412 WA= .913	R2= 0.616 WA= 1.552	R2= 0.826 WA= 2.157	R2= 0.930 WA= 3.523	R2= 1.000 WA=11.102	R2= 1.000 WA=19.683
	AV K=1 K=3 K=7	W1= .159 W3= .288 W7= 2.023	W1= .123 W3= .545 W7= 4.139	W1= .165 W3= 1.309 W7= 4.213	W1= 0.243 W3= 2.059 W7= 8.800	W1= .326 W3= 2.152 W7= 8.613	W1= .373 W3= 4.146 W7=80.435	W1= .380 W3=14.830 W7=-----

SYMBOLS USED

R1 = THEORETICAL VALUE OF CPU UTILIZATION

R2 = EXPERIMENTAL VALUE OF CPU UTILIZATION

WA = AVERAGE WAITING TIME

WK = AVERAGE WAITING TIME FOR THE UNITS NEEDING K QUANTA

Q = QUANTUM SIZE = .5 SEC

TIME REQUIRED =1.0 SEC

FCFS = FIRST COME FIRST SERVED

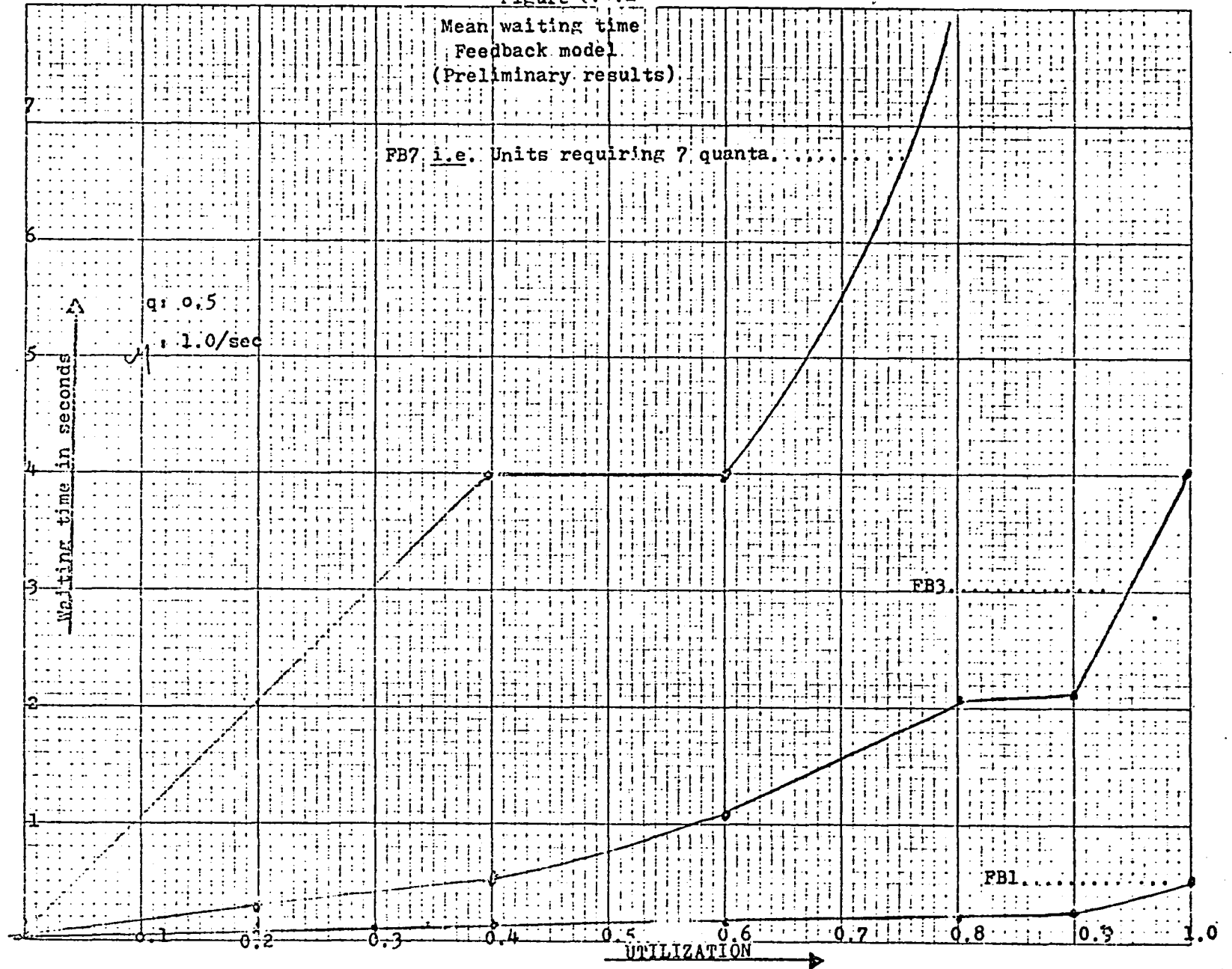
RR = ROUND ROBIN

SJF = SHORTEST JOB FIRST

FB = FEEDBACK OR FOREGROUND BACKGROUND WITH MANY LEVELS OF QUEUES

SOURCE: SUMMARY OF OUTPUT OF THE PROGRAM TS50 LISTED IN APPENDIX TWO
(Same as table 6.2.1)

Figure 6.4.1



Testing the First Hypothesis. In order to investigate further the model i.e. to test the hypotheses, 12 runs of 200 steady state arrivals were made with the program TS50. Table 6.4.2 shows the mean waiting time for units requiring 1, 3, and 7 quanta. Tables 6.4.3 to 6.4.5 show the 95% and the 99% confidence intervals for each type of unit. Table 6.4.8 shows a one-sided confidence limit at 95% and 99%. A graphical representation is given in figure 6.4.2.

In the figure, FB_n indicates the expected waiting time for a unit requiring exactly n quanta i.e. n passes in the system. The graph illustrates three groups of requests i.e. FB_1 , FB_3 , and FB_7 as we did in figure 6.3.2. FB_7 indicates the units requiring 7 quanta, i.e. units needing between 3.000 and 3.5 seconds of service in a model with a fixed quantum of 0.5 second. By comparing figure 6.4.2 with figure 6.3.2 and figure 6.4.3 with figure 6.3.2 one can see whether or not the analytical results fall within the 95% confidence interval obtained from the simulation. By looking at the FB_1 curve in both graphs one can see that the behavior of the units requiring less than one quantum is not the same. For $\rho > 0.8$ the mean waiting time from the analytical solution is significantly larger than the results of the simulation. This can also be observed in table 6.4.8 which gives a one sided 95% to 99% confidence limit.

We then conclude that for a CPU utilization greater than 0.8 the FB_1 results obtained from analytical techniques are too high. We fail to accept the first hypothesis for that area. In a feedback model, a unit requiring only one quantum will have a very short waiting time even though the units requiring many quanta may have an infinite waiting time when ρ is close to 1.0.

WAITING TIME IN THE QUEUES FOR THE FEED BACK MODEL WITH INFINITE NUMBER OF QUEUES																
CPU UTILIZATION ANAL. SIMULATION		UNITS REQUIRING ONE QUANTUM FB1				UNITS REQUIRING THREE QUANTA FB3				UNITS REQUIRING SEVEN QUANTA FB7						
.2	.230	.209	.193	.201	54	52	45	41	366	433	332	255	1039	1118	1320	1060
	.190	.212	.199	.171	37	59	44	31	176	353	258	136	1039	1118	1320	1060
	.165	.201	.195	.199	41	63	45	54	179	382	254	374	794	819	766	801
.3	.345	.314	.239	.302	94	74	82	69	594	591	566	452	1568	1894	3600	6230
	.285	.317	.299	.257	56	94	70	50	314	604	469	320	1084	2776	4180	1132
	.248	.302	.293	.301	59	83	68	78	409	556	426	695	1012	1449	1305	1085
.4	.458	.418	.386	.401	143	108	109	109	919	933	877	796	6711	2524	3365	5250
	.380	.423	.399	.344	91	110	95	86	486	812	810	467	2775	1871	2112	1659
	.330	.403	.392	.401	91	132	110	110	715	806	551	1027	1769	2218	1875	2478
.5	.573	.523	.482	.499	197	170	132	143	1488	1367	1050	1478	11883	4536	3878	7070
	.476	.528	.495	.433	107	170	119	129	766	1053	998	776	4541	3972	2356	4111
	.413	.504	.491	.500	130	151	142	149	958	1344	761	1308	4183	3359	2700	2834
.6	.688	.628	.578	.598	246	182	193	186	2834	1739	1622	2172	22242	9744	5519	10883
	.571	.634	.526	.486	174	197	161	145	978	1326	1049	1187	4786	11014	8071	4758
	.605	.591	.602	.601	191	167	205	210	1790	1010	1768	2100	7963	3339	5402	5911
.7	.802	.732	.669	.699	244	248	217	226	5664	2758	2149	3214	28252	12221	9781	12556
	.654	.738	.687	.618	234	267	203	191	1605	2240	1704	1427	4809	29767	6791	8781
	.579	.694	.688	.709	170	239	215	237	1625	2480	1315	2245	7831	12919	4677	11744
.8	.919	.836	.749	.800	339	75	36	242	9020	3610	2624	5097	72717	16257	11998	53962
	.761	.832	.768	.713	277	308	268	220	2714	3964	2591	2329	9206	33582	15808	8920
	.663	.785	.786	.818	200	295	257	287	2142	3573	1804	2665	9115	19663	9507	14451
.9	.996	.943	.827	.876	390	329	279	315	11541	4932	4138	7353	22267	33212	15087	85476
	.859	.920	.895	.820	293	323	328	283	03479	6839	4882	2947	28729	49633	32033	11273
	.745	.873	.886	.927	275	386	322	369	2542	5094	2394	4414	16757	29081	10616	35197
1.0	1.0	.986	.903	.910	410	364	290	334	19704	6588	5873	10294	158529	104493	18679	100885
	.915	.976	.975	.931	337	372	378	348	04627	1201	6003	3952	66841	117717	47064	19158
	.945	.958	.987	.999	287	408	380	412	3657	7160	3872	7822	24060	74349	49138	78047

NOTE: EACH VALUE SHOWN IN THIS TABLE IS THE MEAN WAITING OF 200 UNITS.
THE TIME IS EXPRESSED IN MILLISECONDS.

TABLE: 6.4.2

SOURCE: SUMMARY OF WAITING TIME OBTAINED BY 96 RUNS OF THE PROGRAM TSSO
WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX TWO.

ARRIVAL DISTRIBUTION : POISSON ; MEAN ARRIVAL RATE = VARIABLE .

REQUIRED SERVICE : EXPONENTIAL WITH MEAN 1.0 SEC. ; QUANTUM SIZE = 5 SEC.

CONFIDENCE LIMITS FOR WAITING TIME MODEL FB UNITS REQUIRING ONE QUANTUM			
CPU R	MEAN OF .12 RUNS	INTERVAL	
		95%	99%
.2	47.2	41.2 - 53.1	38.7 - 55.5
.3	72.8	63.7 - 81.9	60.0 - 85.6
.4	107.8	97.3 - 118.4	92.9 - 122.7
.5	144.9	129.1 - 160.6	122.7 - 167.1
.6	188.1	171.5 - 204.6	164.7 - 211.4
.7	224.2	207.4 - 241.3	200.3 - 248.1
.8	267.0	242.3 - 291.7	232.3 - 301.7
.9	323.5	299.1 - 347.9	289.2 - 351.8
1.0	360.2	333.3 - 387.0	322.4 - 399.9

TABLE: 6.4.3

SOURCE: SUMMARY OF THE OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX SEVEN.

CONFIDENCE LIMITS FOR WAITING TIME MODEL FB3 UNITS REQUIRING 3 QUANTA			
CPU	MEAN	INTERVAL	
		95%	99%
.2	291.5	230 - 352.	206 - 376
.3	499.7	423 - 575	392 - 606
.4	766.5	652 - 880	606 - 927
.5	1112.2	937 - 1286	866 - 1350
.6	1635.2	1273 - 1996	1126 - 2144
.7	2368.8	1618 - 3119	1372 - 3425
.8	3527.3	2282 - 4772	1775 - 5229
.9	5087.9	3468 - 6707	2809 - 7366
1.0	7630.6	4718 - 9543	3531 - 11729

TABLE: 6.4.4

SOURCE: SUMMARY OF THE OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX SEVEN.

CONFIDENCE LIMITS FOR WAITING TIME				
MODEL F37 UNITS REQUIRING 7 QUANTA				
CPU	MEAN	INTERVAL		
		95%		99%
.2	1482	198 - 2726		315 - 3240
.3	2276	1240 - 3311		818 - 3733
.4	2883	1896 - 3876		1494 - 4273
.5	4618	2944 - 6262		2305 - 6931
.6	8288	5074 - 11501		3766 - 12810
.7	12685	7756 - 17615		5748 - 19623
.8	22257	9942 - 35922		4650 - 41213
.9	38959	17864 - 60044		9274 - 68635
1.0	72329	44337 - 100321		32935 - 111723

TABLE: 6.4.6

SOURCE: SUMMARY OF THE OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX SEVEN.

CONFIDENCE LIMITS ON UTILIZATION				
ANAL. VALUE	SIMULATION MEAN	INTERVAL		
		95%		99%
.2	.197	.186 - .208		.181 - .212
.3	.290	.279 - .312		.273 - .318
.4	.395	.373 - .416		.314 - .424
.5	.494	.466 - .519		.455 - .530
.6	.593	.562 - .624		.549 - .637
.7	.690	.653 - .726		.639 - .740
.8	.789	.746 - .828		.729 - .845
.9	.880	.839 - .921		.822 - .938
1.0	.950	.918 - .980		.905 - .999

TABLE: 6.4.7

SOURCE: SUMMARY OF THE OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX SEVEN.

Our results are intuitively correct, for as we said the queue number one has always priority. Even if the system is overloaded a unit coming to queue 1 will be served before any unit of any other queue. Now if the unit requires less than one quantum it will go out, ^{and} if it requires more than one quantum it will go to a lower priority queue so that we may service another unit in queue 1.

For a unit requiring seven quanta we can see also that the analytical results are outside the confidence intervals. So, to see clearly what happens we prepared the graph 6.4.3 on a different scale. One should recall that there are units requiring more than 7 quanta.

FEEDBACK MODEL

ONE-SIDED CONFIDENCE LIMITS UNITS REQUIRING 1 QUANTUM			
CPU R	MEANS OF 12 RUNS	LIMITS	
		95%	99%
.2	47.2	52.0	54.0
.3	72.8	75.6	83.2
.4	107.8	115.9	119.2
.5	144.9	157.1	163.3
.6	188.1	202.3	208.1
.7	224.1	239.3	245.0
.8	267.0	289.7	296.1
.9	323.5	342.5	347.5
1.0	360.2	381.2	394.9

TABLE: 6.4.8

Figure 6.4.2

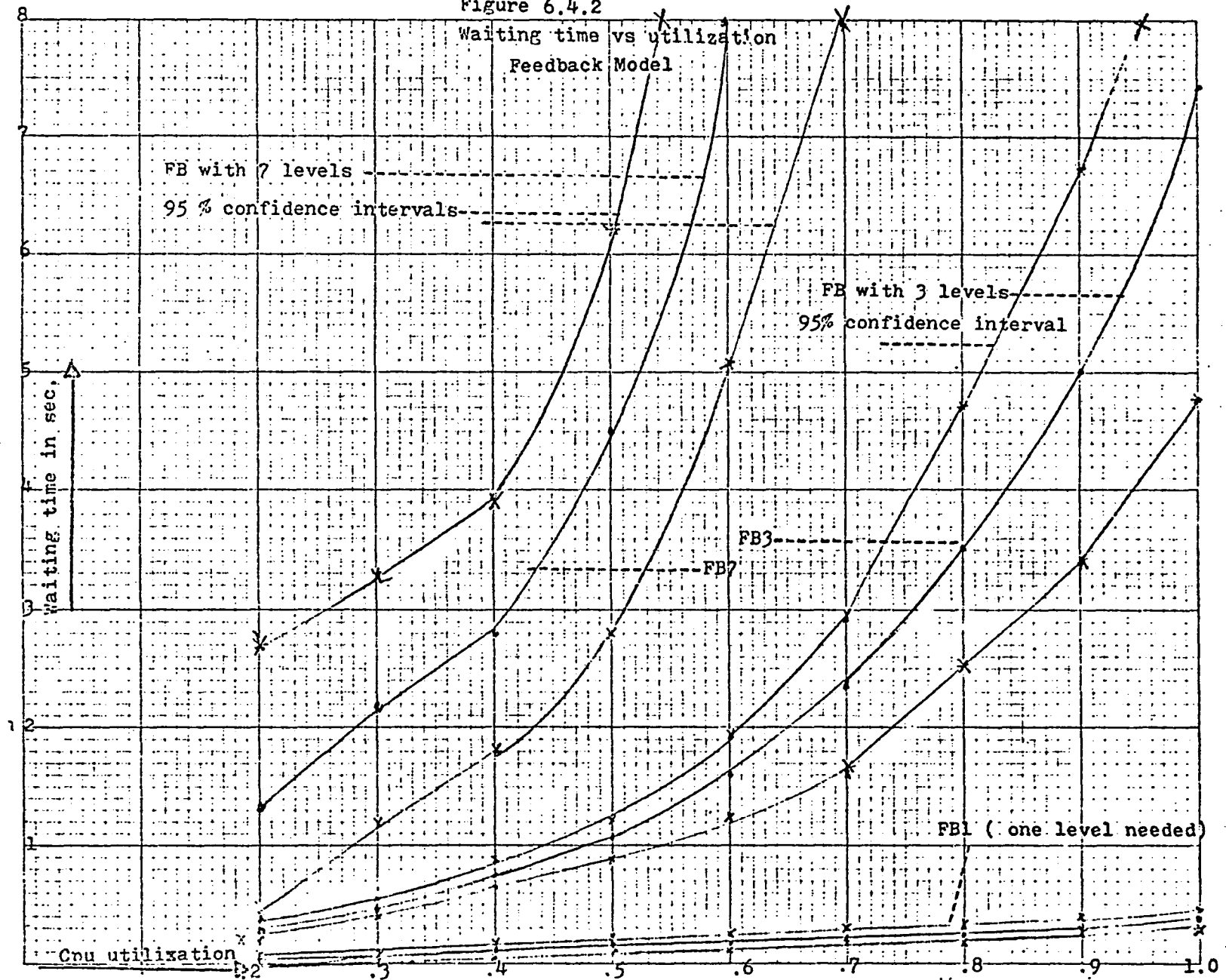
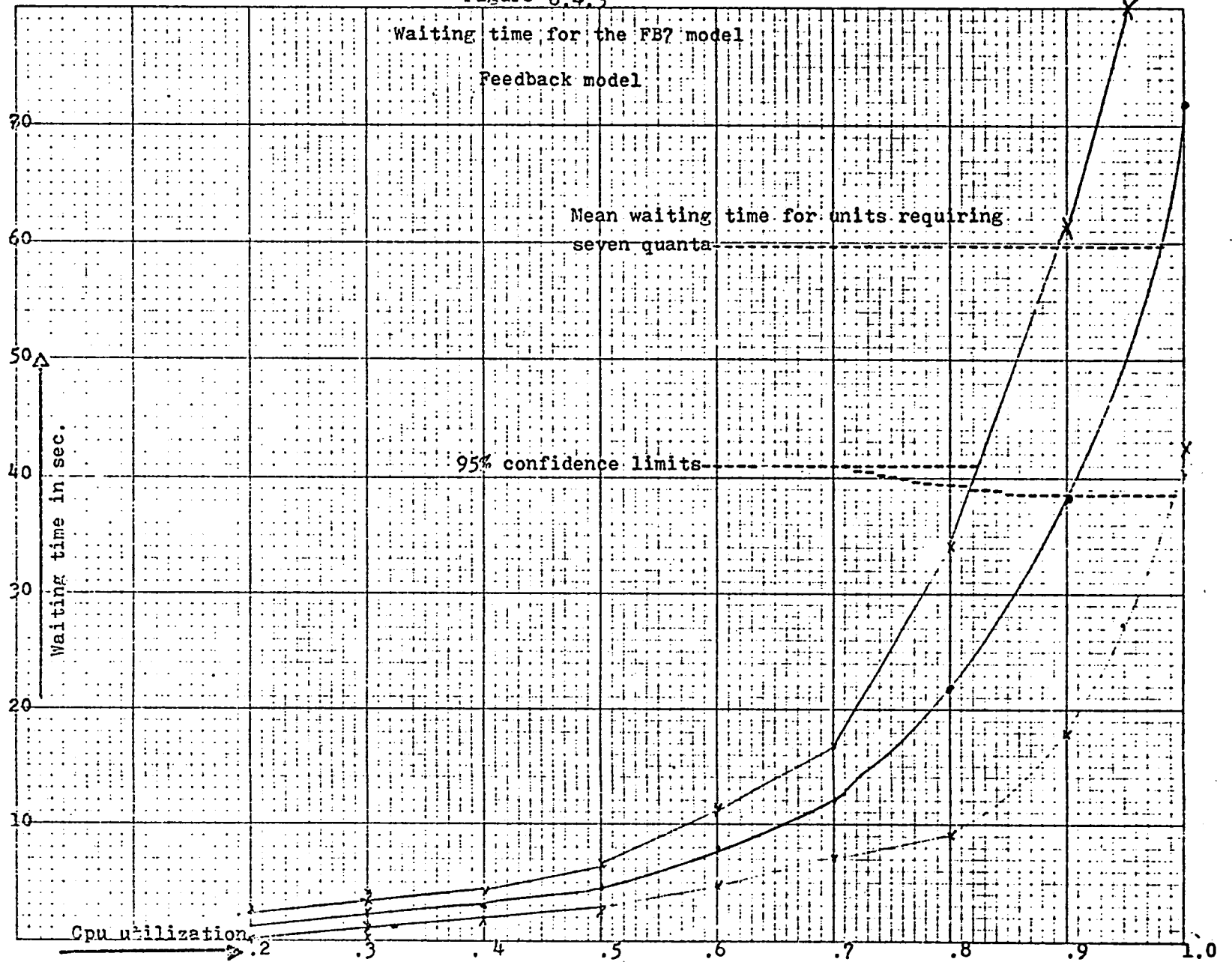


Figure 6.4.3



As shown in figure 6.4.3 and in table 6.4.4 the waiting time for the unit requiring seven quanta for a CPU utilization of 1.0 is between 44 seconds and 100 seconds. This is very long but it is not infinity, as suggested by the formula

Another interesting outcome of our preliminary analysis was that there was a plateau somewhere in the FB curve for both FB7 and FB3. To test this behavior CPU utilization in-between was used. As shown in figure 6.2.4 there is no plateau and it was due only to random sampling.

Because of the expense involved in running simulation models one must select the area of interest rather than run the models for all the points. The author ran the same feedback model with three different distributions of the inter-arrival time. The results were obtained only for high utilization of the CPU as shown in tables 6.4.7 and 6.4.8. In those two tables the response time rather than the waiting time was recorded. In the first case the response time for units requiring more than one quantum was used. In the second case the response time for all units was used.

As shown in tables 6.3.9 and 6.4.10 the arrival pattern has a significant effect on the mean of the response time in both cases. However, there is no significant interaction effect. That means that unlike in chapter five, the CPU utilization, together with the arrival pattern does not have an effect on the response time.

EXPECTED RESPONSE TIME FOR UNITS REQUIRING MANY QUANTA												
CPU UTIL.	POISSON ARRIVAL				UNIFORM ARRIVAL				NORMAL ARRIVAL			
.2	2101	1981	1936	1808	VALUES NOT OBTAINED FOR RC .8							
	1625	1969	1776	1785								
	1636	2001	1888	1905								
.3	2429	2235	2321	2287								
	1827	2242	2087	2083								
	1868	2163	2087	2109								
.4	3571	2603	2688	3222								
	2138	2513	2384	2612								
	2164	2582	2314	2623								
.5	5283	3174	3530	3901								
	2716	3021	2621	3423								
	2624	3006	2685	3001								
.6	8453	4384	3622	4912								
	3192	4461	3247	3927								
	4012	2784	3715	3940								
.7	12187	5384	4333	6346								
	3918	7565	3796	4929								
	3150	5162	5136	5172								
.8	23710	6751	5263	11132								
	5336	10200	5897	5741	2914	3663	1555	4978	3923	4927	2405	5773
	4368	7149	4828	6692	2038	3337	2665	2159	2244	3663	3122	3136
.9	30947	11541	5859	11909	11383	8717	6094	11010	20352	5825	3682	10483
	9351	15449	10335	7268	7068	11069	4158	10940	4684	10538	4131	6069
	6256	9865	6903	10631	7248	6572	5558	6942	6200	5141	4757	7982
1.0	36605	16837	6863	11239	15368	14759	9036	9439	26058	10085	5734	7817
	9273	18980	16198	11505	9355	12742	5462	15232	9238	10385	9704	13132
	10004	14887	12188	14945	10613	12151	9060	6881	8875	10327	9369	8416

NOTE: ALL TIMES ARE EXPRESSED IN MILLISECONDS.
QUANTUM SIZE = 0.5 SECOND.

TABLE: 6.4.7

SOURCE: SUMMARY OF OUTPUT OF 289 RUNS OF THE PROGRAM TS50 LISTED IN
APPENDIX TWO.

EXPECTED RESPONSE TIME FOR ALL UNITS												
CPU	POISSON ARRIVAL				UNIFORM ARRIVAL				NORMAL ARRIVAL			
.2	1384	1248	1227	1232	VALUES NOT OBTAINED FOR R < .7							
	1112	1349	1228	1114								
	1019	1304	1193	1314								
.3	1608	1399	1452	1527								
	1242	1536	1442	1286								
	1159	1406	1320	1456								
.4	2318	1622	1674	2113								
	1455	1709	1636	1602								
	1337	1666	1464	1787								
.5	3378	2180	1964	2548								
	1824	2064	1770	2087								
	1613	1933	1690	2039								
.6	5394	2674	2248	3189								
	2143	3031	2479	2003								
	2514	1857	2505	2671								
.7	7604	3252	2606	4127								
	2624	5031	2539	4929								
	2197	3211	2233	3381								
.8	14676	4177	3088	7177								
	3547	10200	6693	3464	3914	5246	2500	8121	2710	3400	1491	3557
	2587	4346	2946	4406	3396	4847	4524	3066	3663	2446	1876	2179
.9	18351	5811	3383	7576	7000	5220	3568	17062	12545	3461	2218	6699
	6035	9986	6912	4436	4644	7240	2735	6423	3092	6884	2738	3677
	3670	5639	4147	6864	4288	4095	3341	4580	3658	3171	2896	6910
1.0	19699	9583	3887	6746	9008	8776	5202	5976	14908	5897	3320	4715
	5870	11967	10683	7028	6117	7952	3613	8892	05943	6543	4490	7811
	5928	8744	7162	9223	16267	7432	5328	5203	5232	6191	4972	5387

TABLE: 6.4.8

SOURCE: OUTPUT OF 280 RUNS OF THE PROGRAM TSSU LISTED IN APPENDIX TWO.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME FB MODEL
ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	874512384.000	2.	437256192.000	8.900
S(A)	1621286912.000	33.	49129904.000	
B	391122944.000	2.	195561472.000	21.055
AB	9773056.000	4.	2443264.000	0.263
BS(A)	613015552.000	66.	9288114.000	
GRAND MEAN = 8771.645				

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	874530816.000	2.	437265408.000	8.900
S(A)	1621286912.000	33.	49129904.000	
B	391145472.000	2.	195572736.000	21.056
AB	9756674.000	4.	2439168.000	0.263
BS(A)	613015552.000	66.	9288114.000	
GRAND MEAN = 8771.637				

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3
1	8103.914	3299.250	4186.914
2	11326.164	8763.250	7487.164
3	14958.332	10841.500	10678.332

NOTE: ONLY THE UNITS REQUIRING MORE THAN ONE QUANTUM HAVE BEEN INCLUDED.
A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION ; BOTH ARE FIXED.

TABLE: 6.4.9

SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME FB MODEL

ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	135046400.000	2.	67523200.000	4.142
S(A)	538007040.000	33.	16303243.000	
B	136242176.000	2.	68121088.000	12.910
AB	15941376.000	4.	3985344.000	0.755
BS(A)	348268544.000	66.	5276796.000	
GRAND MEAN = 5790.711				

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	135045152.000	2.	67522576.000	4.142
S(A)	538007040.000	33.	16303243.000	
B	136240160.000	2.	68120080.000	12.909
AB	15940611.000	4.	3985152.000	0.755
BS(A)	348268544.000	66.	5276796.000	
GRAND MEAN = 5790.711				

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3
1	5592.250	5053.582	2916.917
2	7400.832	5016.332	4410.750
3	8794.164	6647.164	6284.414

NOTE: ALL UNITS HAVE BEEN INCLUDED WITHOUT REFERENCE TO SERVICE NEEDED.
A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION ; BOTH ARE FIXED.

TABLE: 6.4.10

SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

SHORTEST JOB FIRST

The Model. In this model a unit is serviced until completion. However, when a job joins the queue it is assigned a priority based on the amount of processing needed. In previous time-sharing models as we have seen, the unit having a smaller requirement do better. However, the amount of processing needed is not known a priori. One may what extent of improvement can be obtained by knowing in advance the exact service requirement.

Parameters and Assumptions. The following assumptions are made:

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad 6.1.01$$

where the mean inter-arrival time is $1/\lambda$ seconds and the mean arrival rate is λ , units per second.

2. For the service process an exponential service distribution is assumed

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad 6.1.02$$

where the mean service requirement is $1/\mu$ seconds.

3. The quantum is q
4. There is exact knowledge in advance of the required processing time t .

5. There is an infinite supply of customers.
6. The measure of efficiency is the expected waiting time conditional on the service requirement.
7. There is no time lost for setting up, i.e. an interrupted unit carries with it the knowledge of the processing already done to it.
8. When the service facility completes the service of a unit it examines the queue at that time and determines the unit with the shortest requirement.
9. There is no preemption, i.e. a unit requiring a long service time is processed to completion even though a shorter job might come during that time.

Analytical Solution. In order to solve the model one can use the job size to assign external priorities. In our case, priorities are continuous because they are assigned by the time (t) i.e. the job requirement which is distributed according to an exponential distribution. Once the priorities have been obtained, it becomes a model with external continuous priorities. The following results have been obtained by Phipps¹⁸ who used continuous priorities rather than discrete priorities as done by Cobham¹⁹. The waiting time for a unit requiring t seconds of service is:

$$W(t) = \frac{p(1/\mu)}{1 - \lambda/\mu [1 - e^{-\mu t} (1 - \lambda t)]^2} \quad 6.5.1$$

¹⁸ Thomas E. Phipps, "Machine Repair as a Priority Waiting-Line Problems," Operations Research, January 1956, Vol. 4, pp. 76-86

¹⁹ Cobham, "Priority Assignment in Waiting Line Problems," Operations Research, Vol. 2, (February 1954), pp. 70 - 76.

The expected number of jobs in the waiting line in the unsaturated is given by

$$N = \frac{\lambda^2}{\mu} \int_0^{\infty} \frac{e^{-\mu t} dt}{\{1 - (\lambda/\mu) [1 - e^{-\mu t} (1 + \mu t)]\}^2} \quad 6.5.2$$

Using the Pollaczek-Kintchine formula the average waiting time can be obtained as

$$E(W) = \frac{N}{\lambda} \quad 20 \quad 6.5.3$$

Therefore the following result can be obtained

$$E(W) = \frac{\lambda}{\mu} \int_0^{\infty} \frac{e^{-\mu t} dt}{\{1 - (\lambda/\mu) [1 - e^{-\mu t} (1 + \mu t)]\}^2} \quad 6.5.4$$

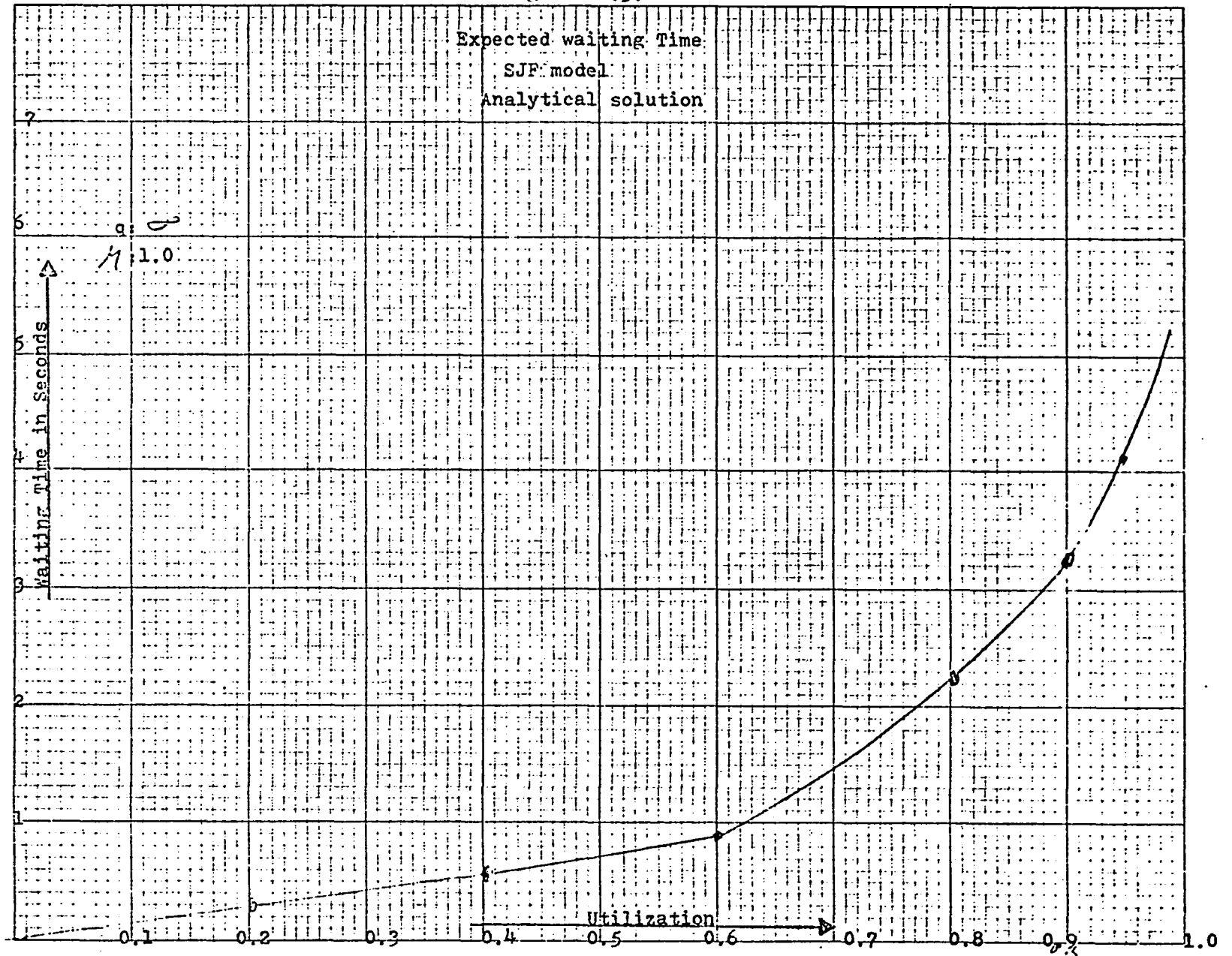
Replacing values in the above formula one may obtain the graph shown in figure 6.5.1.

In the preceding solution the arrival pattern is modified by priority assignments. One may want to determine whether the model is still valid for other than Poisson arrivals. In this section we will test the arrival distribution. In the next section we will study the SJF models with reference to the other models.

Hypotheses.

- a. The mean waiting time is properly obtained from the analytical solution.
- b. The arrival distribution does not have a significant effect on the mean response time.

Figure 6.5.1



Simulation Results. As discussed above, many preliminary runs were made. The results are given in table 6.6.1. A graph of the results is given in figure 6.6.1. We already see that the mean waiting time is less than in the other cases. We will discuss that aspect in the next section. In order to investigate further, the author ran 12 independent runs of 200 steady state units with some high values of utilization. A confidence interval was built around the mean waiting time and it is shown in figure 6.6.2. We see that the mean waiting time obtained from the analytical techniques falls within the 95% confidence interval of the mean waiting time obtained from simulation experiments.

We do not have enough evidence to reject the hypothesis. The analytical solution cannot be rejected. The author also tested the effect of the arrival pattern on the response time. An analysis of variance was performed on the response times for the Poisson, the Uniform, and the Normal distributions. The results are summarized in tables 6.6.2, 6.6.3, 6.6.4 and 6.6.5. The results indicate that the arrival distribution does have an effect on both the waiting time and the response time but that the interaction of the CPU utilization and the arrival pattern has no effect.

COMPARISON OF QUEUE DISCIPLINES

Having studied different queuing disciplines and having tested both the results of the analytical solution, and also the effect of a departure from "Poisson" one may wish to compare the different disciplines in terms of the mean waiting time.

Figure 6.6.1

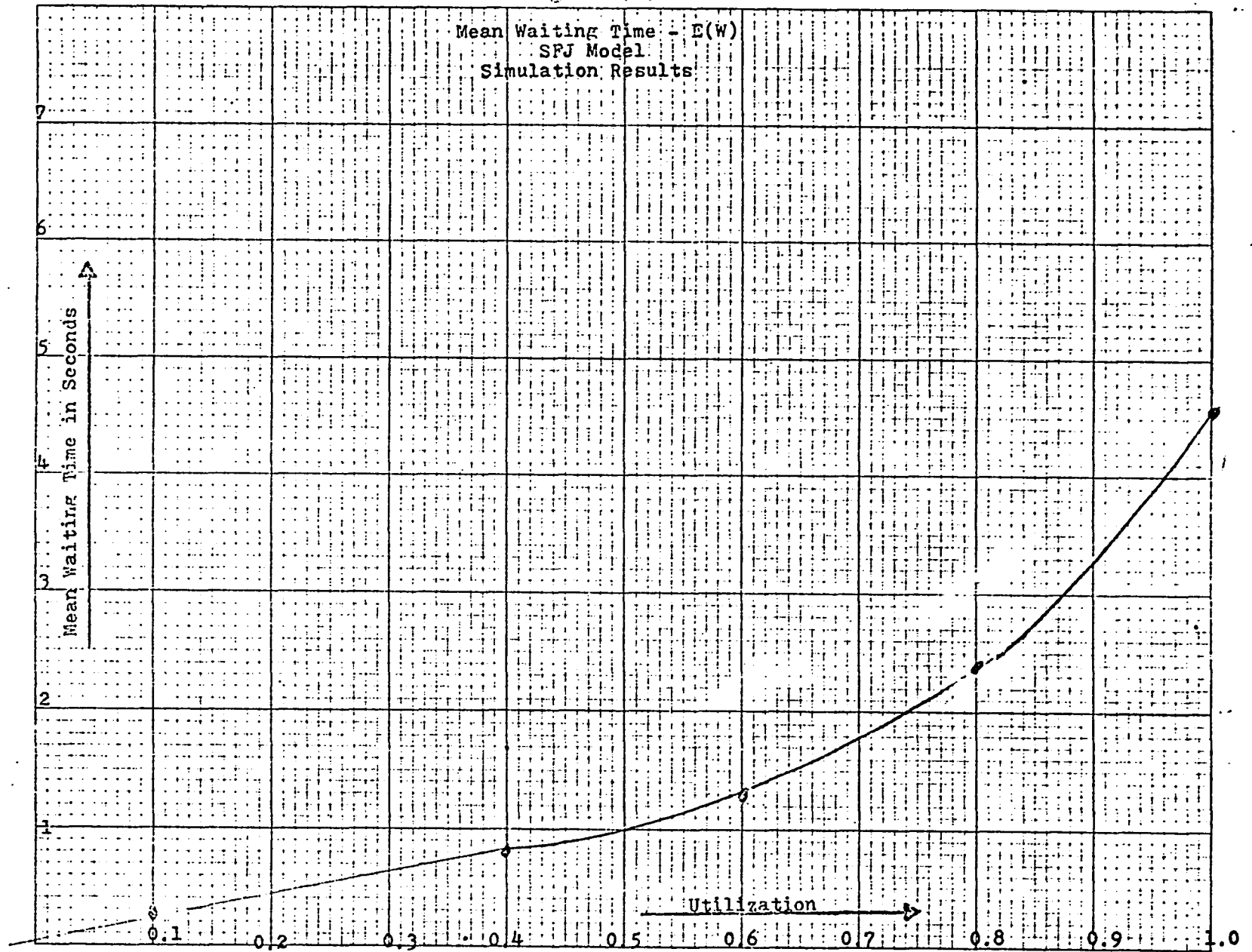


TABLE 6.6.1
WAITING TIME FOR DIFFERENT QUEUE DISCIPLINES

		R1= 0.2	R1= 0.4	R1= 0.6	R1= 0.8	R1= 0.9	R1= 1.0	OVER LOAD
	FCFS	R2= 0.209 WA= .187	R2= 0.425 WA= .572	R2= 0.646 WA= 1.054	R2= 0.861 WA= 2.631	R2= 0.963 WA= 4.015	R2= 1.000 WA=12.508	R2= 1.000 WA=27.667
	FCFS K=1 K=3 K=7	R2= 0.201 W1= .029 W3= .159 W7= .175	R2= 0.406 W1= .165 W3= .509 W7= .334	R2= 0.615 W1= .375 W3= .845 W7= 1.172	R2= 0.827 W1= .897 W3= 1.232 W7= .736	R2= 0.935 W1= 1.586 W3= 1.606 W7= 1.429	R2=1.000 W1= 3.873 W3= 4.683 W7= 4.455	R2= 1.000 W1=16.917 W3=20.783 W7=19.584
	RR AV K=1 K=3 K=7	R2= 0.136 WA= .215 W1= .055 W3= .299 W7= .670	R2= 0.374 WA= .462 W1= .159 W3= .662 W7= 1.371	R2= 0.561 WA= .792 W1= .231 W3= 1.073 W7= 1.958	R2= 0.756 WA= 1.835 W1= .637 W3= 2.153 W7= 5.651	R2= 0.973 WA= 4.800 W1= 1.920 W3= 5.705 W7=14.875	R2= 0.941 WA= 9.907 W1= 4.016 W3=12.241 W7=25.690	R2= 1.000 WA=21.818 W1= 7.622 W3=23.446 W7=53.689
	SJF	R2= 0.222 WA= .435	R2= 0.447 WA= .722	R2= 0.674 WA= 1.313	R2= 0.895 WA= 2.004	R2= 0.952 WA= 3.069	R2= 1.000 WA=4.735	R2= 1.000 WA= 6.957
	FB AV K=1 K=3 K=7	R2= 0.206 WA= .328 W1= .159 W3= .288 W7= 2.023	R2= 0.412 WA= .913 W1= .123 W3= .545 W7= 4.139	R2= 0.616 WA= 1.552 W1= .165 W3= 1.309 W7= 4.213	R2= 0.826 WA= 2.157 W1= 0.243 W3= 2.059 W7= 8.800	R2= 0.930 WA= 3.523 W1= .326 W3= 2.152 W7= 8.613	R2= 1.000 WA=11.102 W1= .373 W3= 4.146 W7=80.435	R2= 1.000 WA=19.683 W1= .380 W3=14.320 W7=-----

SYMBOLS USED

R1 = THEORETICAL VALUE OF CPU UTILIZATION

R2 = EXPERIMENTAL VALUE OF CPU UTILIZATION

WA = AVERAGE WAITING TIME

WK = AVERAGE WAITING TIME FOR THE UNITS NEEDING K QUANTA

Q = QUANTUM SIZE = .5 SEC

TIME REQUIRED =1.0 SEC

FCFS = FIRST COME FIRST SERVED

RR = ROUND ROBIN

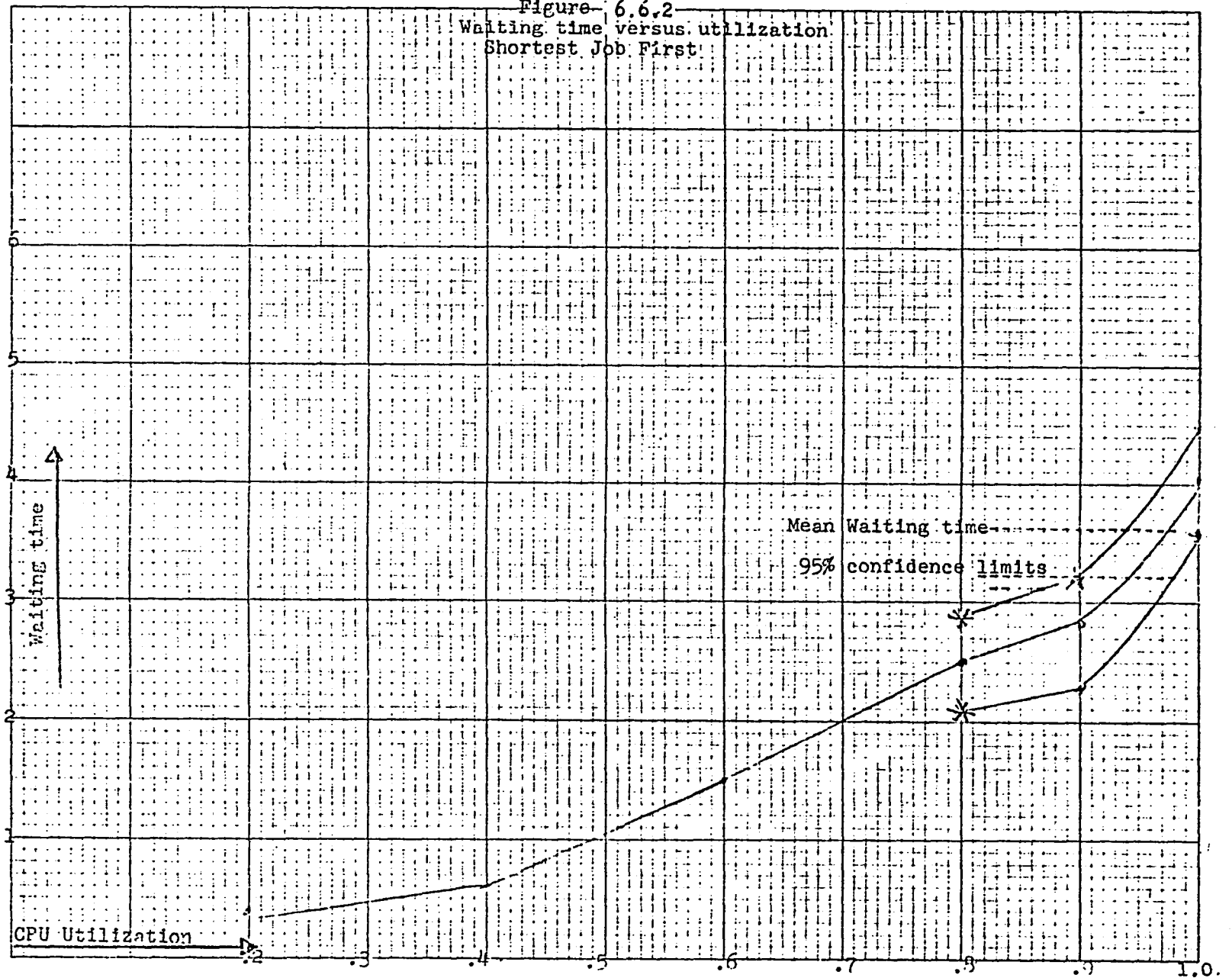
SJF = SHORTEST JOB FIRST

FB = FEEDBACK OR FOREGROUND BACKGROUND WITH MANY LEVELS OF QUEUES

SOURCE: SUMMARY OF OUTPUT OF THE PROGRAM TS50 LISTED IN APPENDIX TWO

Same as table 6.2.1., repeated here for the reader's convenience

Figure 6.6.2
Waiting time versus utilization
Shortest Job First



SUMMARY OF RESPONSE TIME SHORTEST JOB FIRST													
CPU UTILIZATION		POISSON ARRIVAL				UNIFORM ARRIVAL				NORMAL ARRIVAL			
0.8		3200	3758	1731	2233	3355	3316	1930	2309	3292	2498	1564	1978
		2543	3085	2048	2929	1908	2446	1672	3385	1809	2486	1661	1260
		1629	2938	2051	2548	1564	2163	1097	1798	2204	1632	2347	1844
0.9		2732	4117	1831	2182	4073	4937	1823	2534	2320	2972	2111	2157
		2521	3335	2724	3591	2665	3126	1864	2520	1982	4850	2733	3330
		1897	2766	3398	2183	2010	1741	3373	2123	1636	2855	2182	2151
1.0		4210	4456	2695	4015	4476	4600	3828	3808	4108	3851	2066	1890
		3603	5560	4205	4134	3645	4432	2727	4681	2448	2336	2451	3580
		2934	4329	4103	4830	4767	3471	4304	3942	2699	3488	3036	2480

TABLE: 6.6.2

SOURCE: OUTPUT OF 96 RUNS OF PROGRAM TS50 LISTED IN APPENDIX TWO.

SUMMARY OF WAITING TIME SHORTEST JOB FIRST													
CPU UTILIZATION		POISSON ARRIVAL				UNIFORM ARRIVAL				NORMAL ARRIVAL			
0.8		1199	2702	938	1748	2086	2041	1047	1440	2251	1444	690	1111
		1556	2578	1060	1756	898	1379	750	2184	844	1422	716	1515
		769	1925	1044	1358	692	1121	1097	911	894	1157	1086	1405
0.9		1670	3557	1055	1747	2975	3588	1298	1934	3897	2589	1008	1483
		1859	3726	1606	2314	1821	2593	926	3736	1197	2258	1401	2999
		1917	2740	1740	2153	1062	1741	1402	1545	689	1804	1549	1388
1.0		3599	4545	2100	4806	4635	3518	1828	3588	5685	3296	1247	1721
		3147	5174	3174	2793	2889	4010	1729	4273	1667	2873	1476	2900
		1853	3513	3099	3721	2018	3338	2251	2268	1334	2718	2024	1645

663

TABLE: 6.6.3

SOURCE: OUTPUT OF 96 RUNS OF PROGRAM TS50 LISTED IN APPENDIX TWO.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME SJF MODEL

ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	37857536.000	2.	18928768.000	22.684
S(A)	27536640.000	33.	834443.625	
B	7070720.000	2.	3535360.000	8.224
AB	3328000.000	4.	832000.000	1.935
BS(A)	28372736.000	66.	429889.938	
GRAND MEAN = 2924.203				

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	37857984.000	2.	18928992.000	22.685
S(A)	27536640.000	33.	834443.625	
B	7071361.000	2.	3535680.000	8.225
AB	3327936.000	4.	831984.000	1.935
BS(A)	28372736.000	66.	429889.938	
GRAND MEAN = 2924.202				

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3
1	4089.500	4046.750	3036.583
2	2772.583	2832.500	2608.933
3	2556.917	2325.500	2048.567

NOTE:

A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION : BOTH ARE FIXED.

TABLE: 6.6.4

SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE WAITING TIME SJF MODEL

ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	45598208.000	2.	22799104.000	14.021
S(A)	53658624.000	33.	1626018.000	
B	5740288.000	2.	2870144.000	8.440
AB	2300160.000	4.	575040.000	1.691
BS(A)	22444800.000	66.	340072.688	
GRAND MEAN = 2092.055				

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	45598656.000	2.	22799328.000	14.022
S(A)	53658624.000	33.	1626018.000	
B	5740609.000	2.	2870304.500	8.440
AB	2299968.000	4.	574992.000	1.691
BS(A)	22444800.000	66.	340072.688	
GRAND MEAN = 2092.053				

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3
1	3454.500	2954.583	2399.667
2	2098.667	2051.750	1802.500
3	1552.583	1303.833	1210.417

NOTE:
TABLE: 6.6.5A = CPU UTILIZATION B = ARRIVAL DISTRIBUTION : BOTH ARE FIXED.
SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

Analytical Results. In the previous sections we have obtained results for each of the queue disciplines considered, i.e. Round-Robin, Feedback, and Shortest Job First. Using the results obtained above we may compare them with each other. and with the classical First Come First Served.

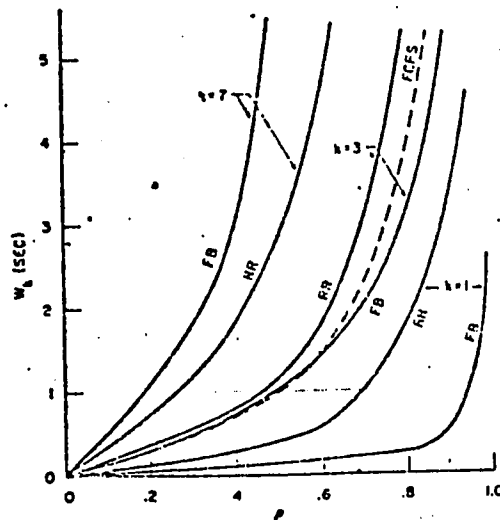


Figure 6.7.3 Comparison of ∞ -level FB and RR conditional waiting times

Source: E. Coffman and L. Kleinrock, "Feedback Queuing Model for Time Shared Systems," Journal of ACM, Vol. 15, No. 4, (October 1968), p. 562-563.

Figure 6 7.1 shows the relationship between the FB and RR model as obtained in the analytical solutions. Figure 6.7.2 shows the relationship between the FCFS and SJF models. It is obvious that FB, RR, and SJF disciplines favor shorter jobs. However, one may wish to make a comparison with the FCFS to see how the discrimination will affect the users. We may recall how the selection of users is made.

1. The SJF discipline discriminates on^{the} basis of future service.
2. The RR and the FB discipline discriminate on^{the} basis of past service.
3. The FCFS does not discriminate.

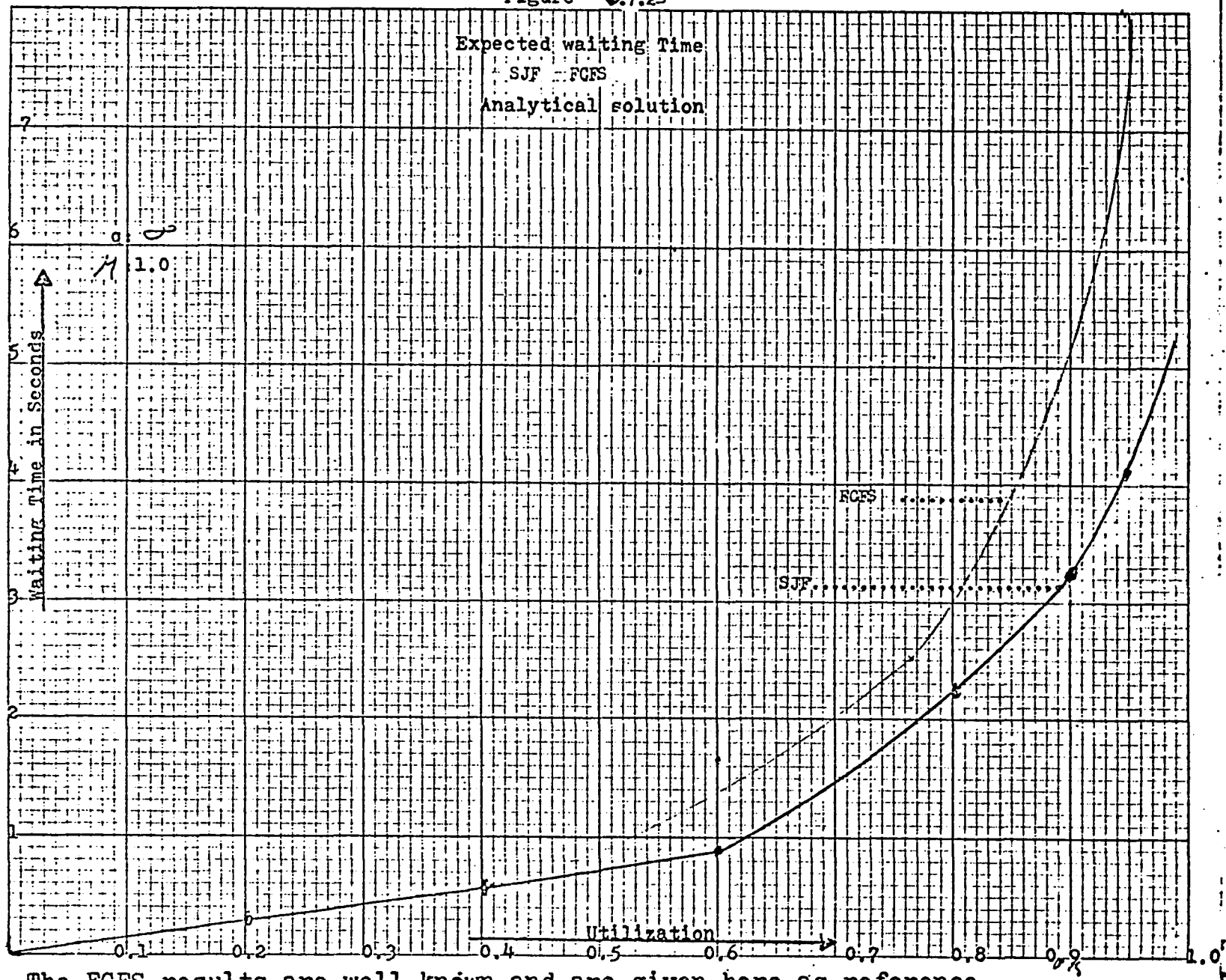
One can see from figure 7.7.1 and 7.7.2 that:

1. A unit requiring 1 quantum does better in the FB i.e the expected waiting time is less
2. The unit with shorter service demand does better in RR and FB than in FSFC.
3. The unit requiring 7 quanta does better in RR than in FB.
4. In general a unit does better in SJF than in any of the queue disciplines.

Hypotheses. In the light of these results we will test the following hypotheses:

- a. For the units requiring one quantum the Feedback model is more favorable than the Round Robin and the FCFS.

Figure 6.7.2



The FCFS results are well known and are given here as reference

- b. The unit requiring 3 quanta does better in the FB than in the RR, does better in FB than in FCFS, and does better in FCFS than in RR.
- c. The mean waiting time is less in general in the SJF discipline than in FCFS, RR, or FB models.

Simulation Results. In order to test the above hypotheses we will use the data already obtained in previous sections.

relationship could easily be obtained by plotting, for all queue disciplines, the mean waiting time with a 95% confidence interval. However, in order to have more clarity we will present on figures 6.8.1, 6.8.2, and 6.8.3 the mean waiting time on a confidence interval for the RR, FB, and FCFS queuing disciplines respectively. Figure 6.8.4 shows the mean waiting time for all queue disciplines. One can easily see that the relationship arrived at by the analytical solution does hold. If we also add the mean waiting time for ^{the} classical FCFS model to the graph on figure 6.8.3, one can see that the first two hypotheses cannot be rejected because in each case the same relationship applies. Even if the model does not behave exactly as found by analytical solution, the relationship among the queue disciplines RR, FB, and FCFS is the same.

We now wish to verify a claim made in the analytical solution that a customer always does better in a model where the shortest job goes first. Figure 6.8.4 shows the mean waiting time in the case of the FCFS, SJF, FB, and RR models. One can easily see that when ρ is > 0.6 the mean waiting time of the SJF is much less than the mean waiting time in the FCFS.

Figure - 6.8.1

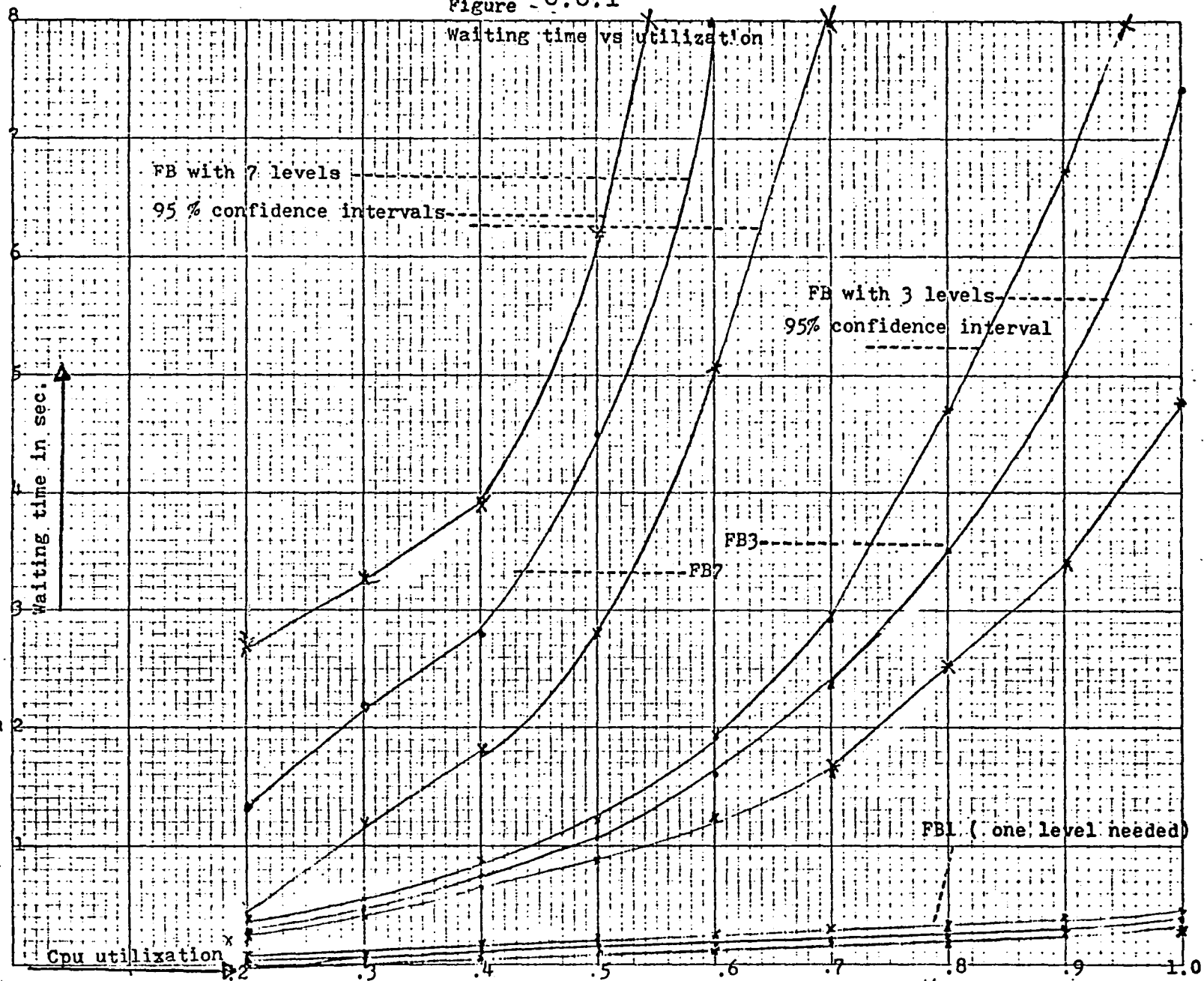


Figure 6.8.2

Waiting time versus Utilization
RoundRobin Model

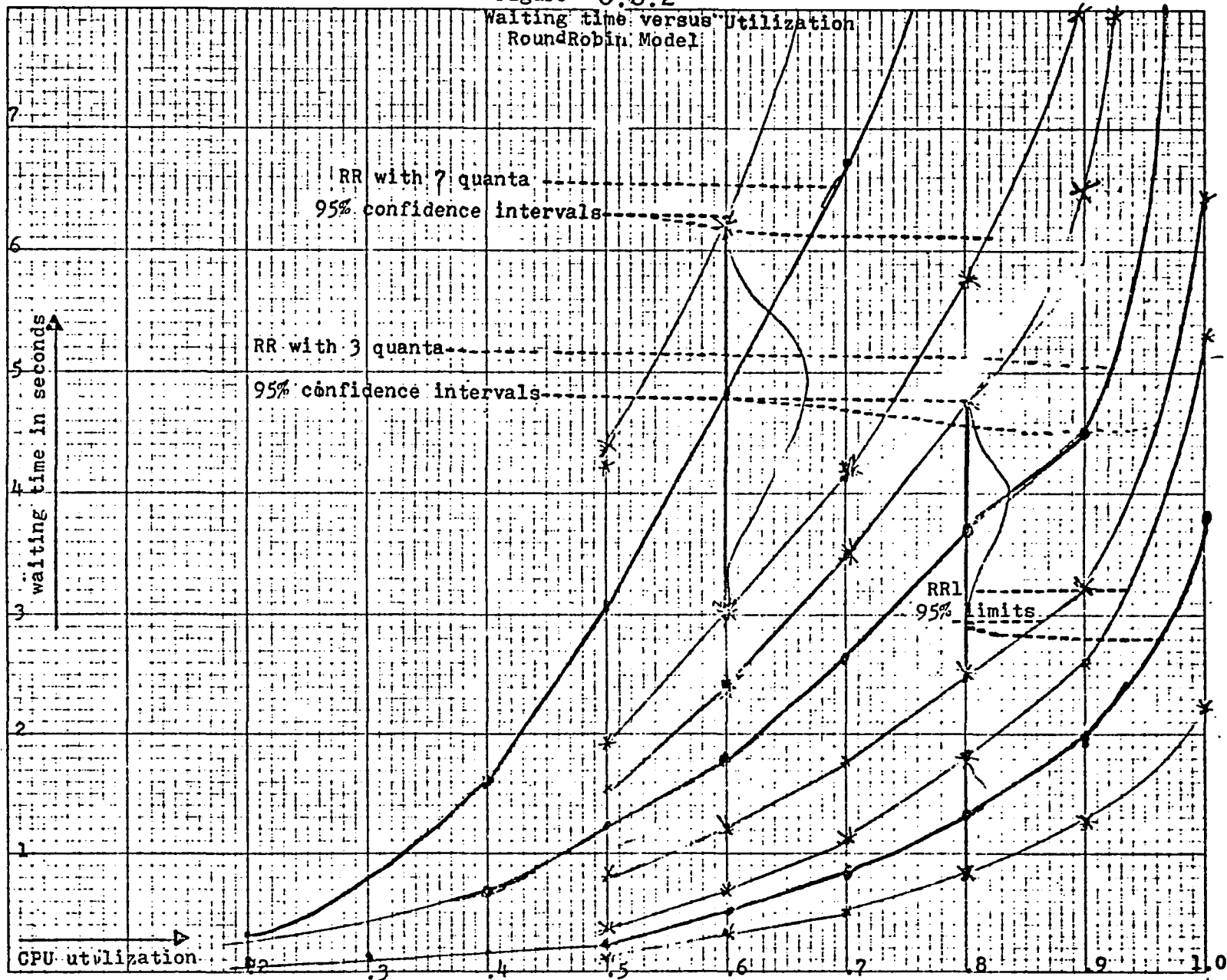


Figure 6.8.3

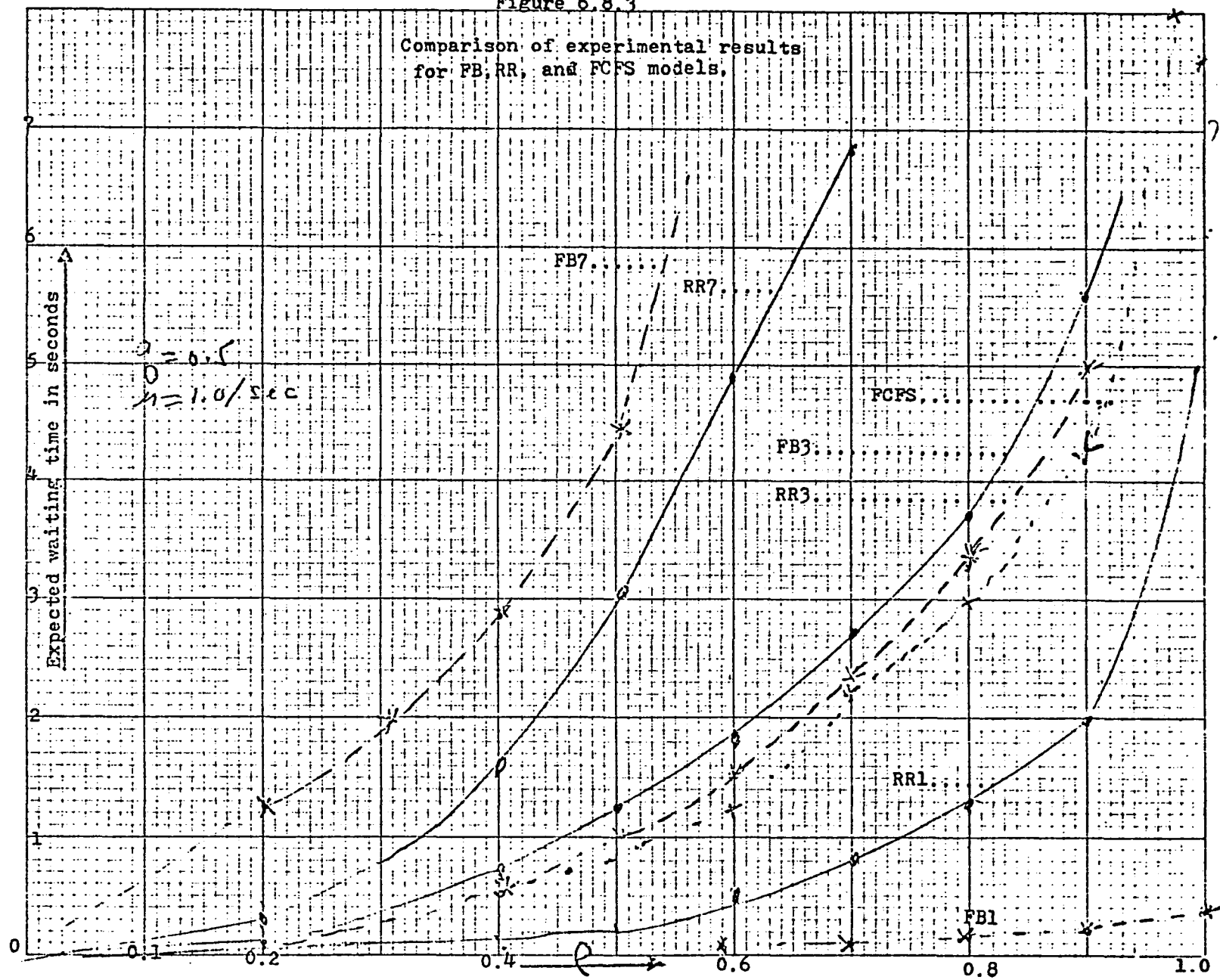


Figure 6.8.4

Waiting time versus Utilization

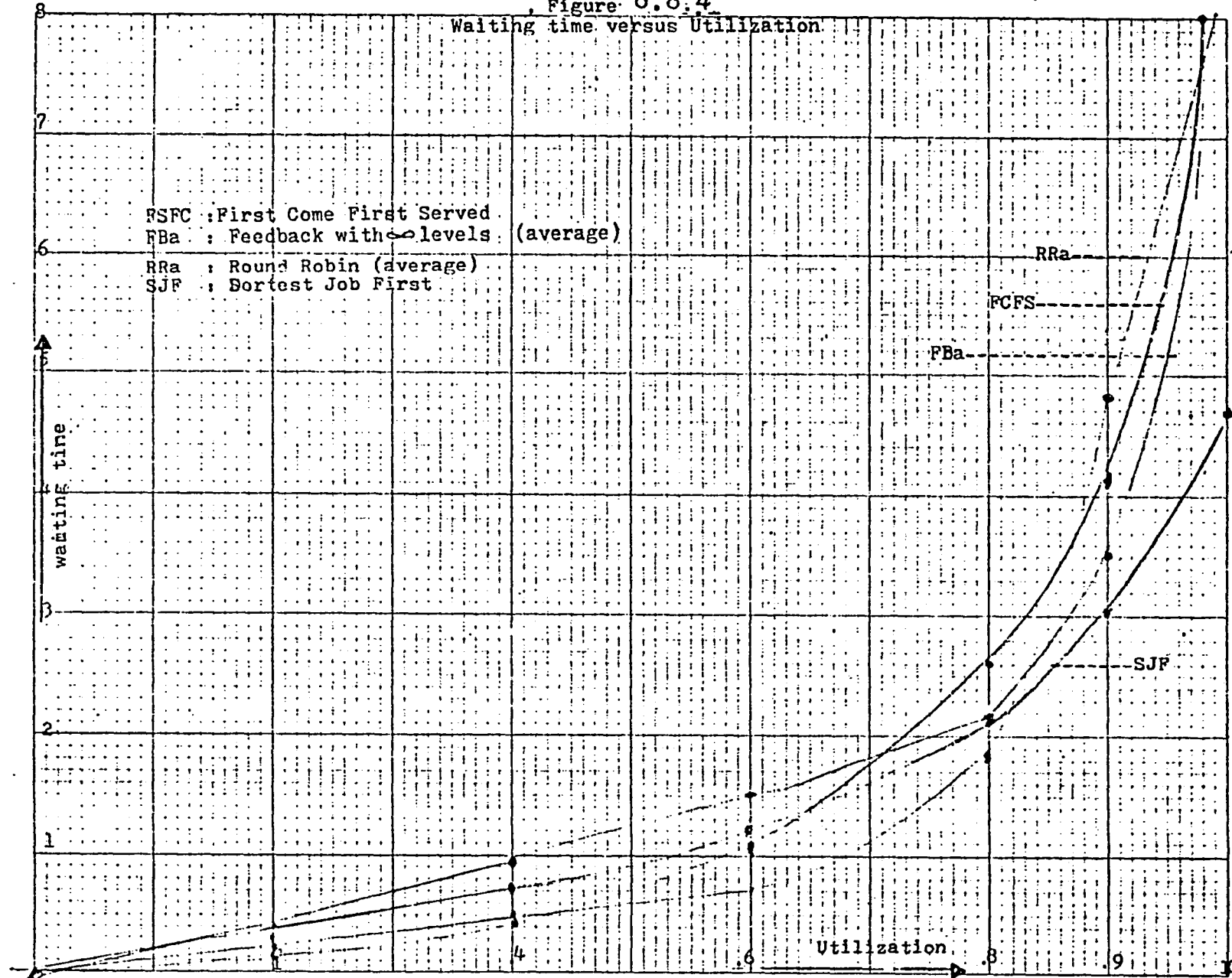
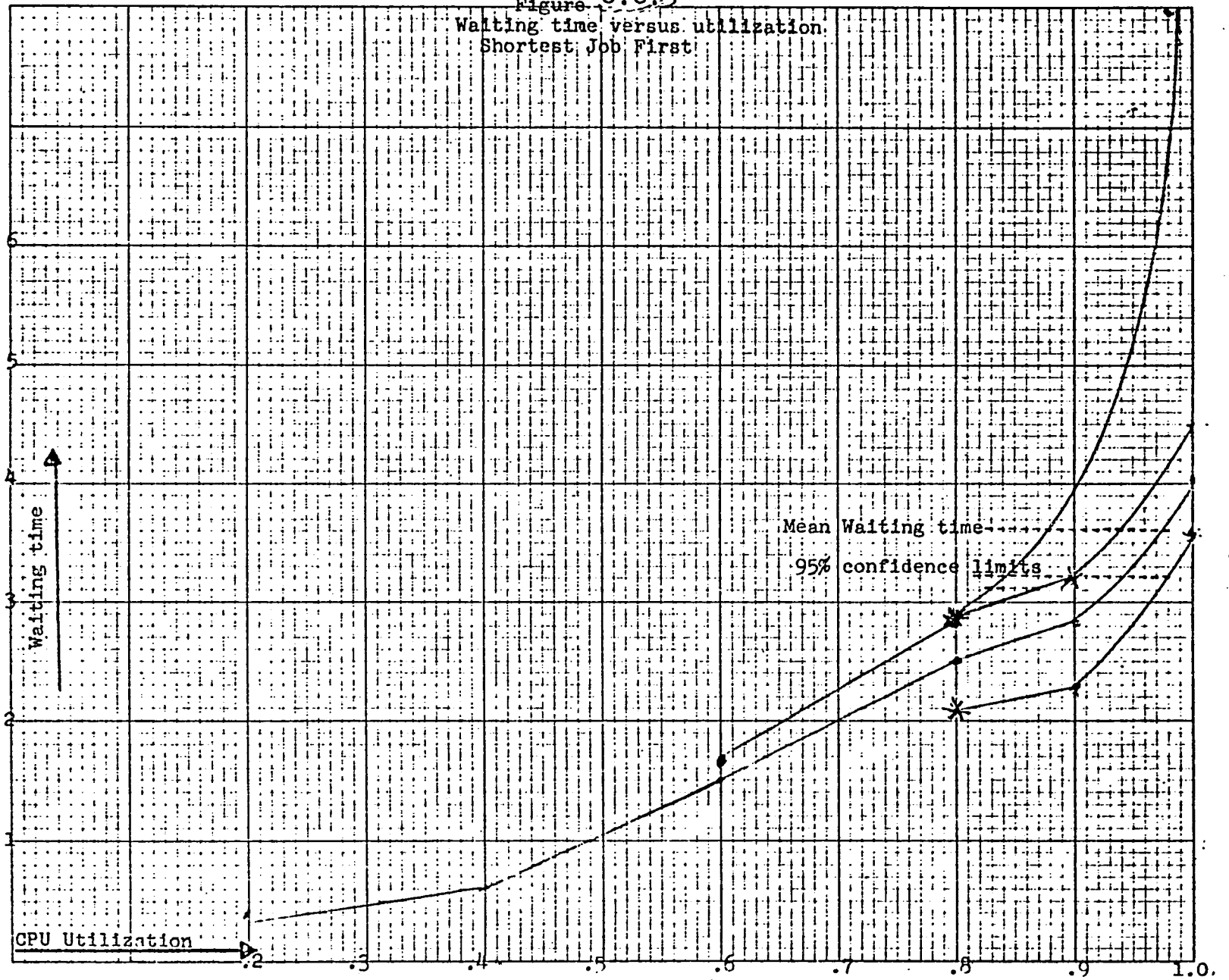


Figure 6.8.5
Waiting time versus utilization.
Shortest Job First



In figure 6.8.5, one can see, by looking at the confidence intervals, that the SJF unit does much better in terms of the mean waiting time. We therefore do not reject the second hypothesis. The relationship expressed in the analytical solution is maintained.

CONCLUSION

In this chapter the author has reviewed and tested a time-sharing model with different queue disciplines. Original results were found concerning the behavior of FB_1 models as the CPU utilization approaches 1. Confidence limits of the expected waiting times were not obtained for every point but only for high utilization of the CPU. Since the preliminary results in other areas were consistent with the analytical results, the author elected to allocate limited computer resources to areas which seemed to depart from expected results. In designing experiments one has to consider the cost of such experiments and get as much reliable information as possible at the minimum cost.

The simulation results were significantly different from the analytical results for high utilization values in the Round Robin and the Feedback models. In the Feedback model, we showed that the unit requiring less than one quantum will have a much shorter mean waiting time than the one proposed in the analytical solution. However, we found that the analytical results for the SJF model were consistent with our experimental results.

We also tested the effect of a departure from the usual Poisson arrival assumption on the different models measuring both the mean response time and the mean waiting time. We showed by analysis of variance that the arrival pattern has a significant effect in all cases but that there is no interaction effect, i.e. the arrival pattern together with the service station utilization have no significant effect.

The results on the effect of the arrival distribution are different from the one obtained in previous chapters. They imply that in the time-sharing models analyzed in this chapter the analytical results obtained under the Poisson assumption are not valid for any type of arrival distribution. However, by simulation, as we have showed, we can obtain the mean response time for any type of arrival distribution.

The general relationship among the different queue disciplines expressed in the analytical solutions was maintained in the simulation experiments. Interesting results were obtained to show that when the "shortest job goes first", the overall mean waiting time is less, so that the user in general does better.

CHAPTER SEVEN

FINITE TIME-SHARING MODELS

The purpose of this chapter is to test the analytical solution of time-sharing queuing models with a finite number of feedback queues. In the previous chapter we introduced the concept of time-sharing systems and we analyzed different queue disciplines such as Round-Robin, Feedback, and Shortest Job First. It was found that in some cases the results of analytical solutions were confirmed by simulation. However, in some other cases they were not. In this chapter, we will analyze time-sharing models with some restrictions. First, a model will be analyzed in which the unrealistic assumptions of an unlimited number of customers and of no over-head are removed. Second, we will analyze a model similar to the feedback model, but where the number of feedback queues is limited.

We will study the models by simulation and will relate them to the models studied in the previous chapter. However, no sensitivity analysis on the arrival distribution will be made. This is because sensitivity analysis has already been made on the similar models in Chapter Six and because we have to allocate limited computer resources. As will be explained later, these models are extremely long to simulate because of the large number of passes required. Therefore, we will use a different pattern for the presentation.

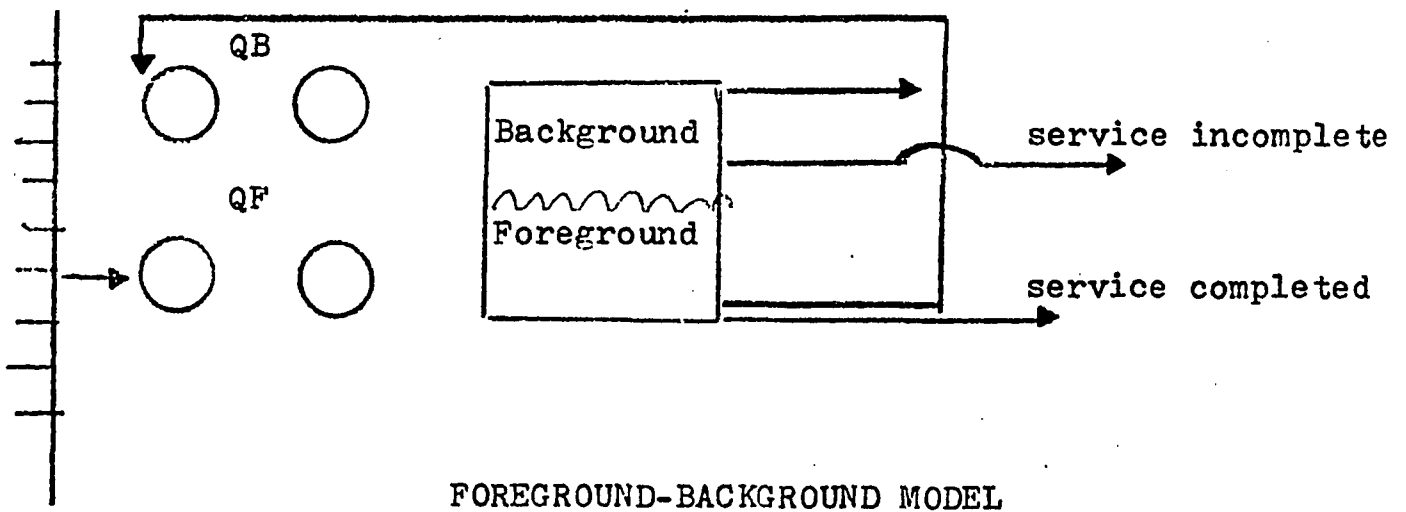
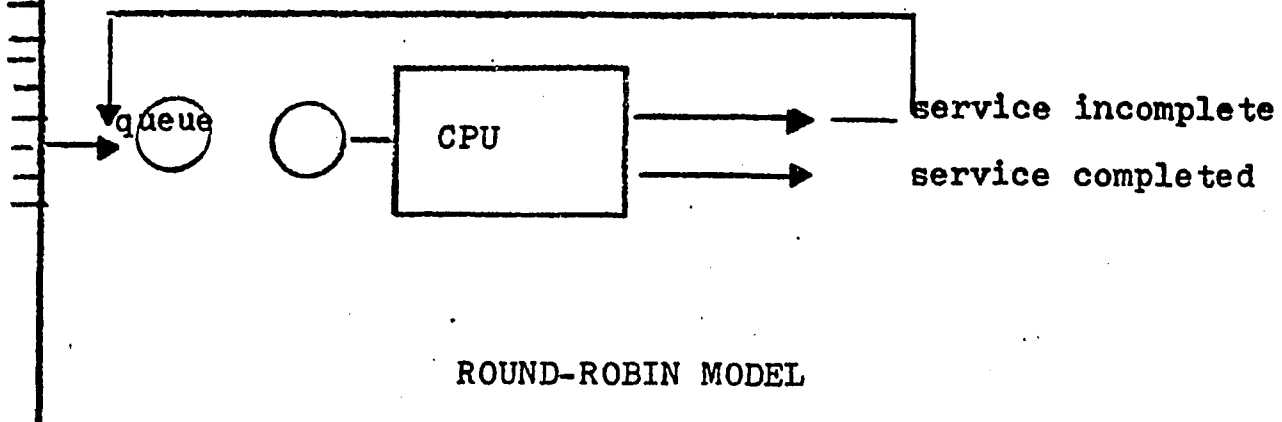


Figure 7.1.1 - Two models for Time-Sharing

Source: J.M. McKinney, "A Survey of Analytical Time-Sharing Models", Computing Survey, Vol. 1, No. 2, (June 1969) p. 106.

FINITE NUMBER OF CUSTOMERS

The Model to be Studied. In this model there is a fixed number of customers, N , who generate requests for a single central processor. The customers are serviced following a Round-Robin queue discipline as described in the previous chapter and reproduced graphically in figure 7.1.1. The customer is given a limited quantum of service; If the processing is not completed at the end of this time, he goes back to the end of a single queue with the arriving units. The interrupted unit keeps the knowledge of the processing done to it. However, unlike in the previous Round-Robin model, there is some time lost by interrupting the processing, i.e. there is a setting-up time or over-head which makes the model more like real life cases.

Parameters and Assumptions. The following assumptions are made.

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad 7.1.01$$

where the mean inter-arrival time is $1/\lambda$ seconds and the mean arrival rate is λ units per second.

2. For the service process an exponential service distribution is assumed,

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad 7.1.02$$

where the mean service requirement is $1/\mu$ seconds.

3. The quantum size is a constant value θ .
4. A fixed overhead is added at the end of each quantum r .
5. There is only one level of queue, i.e. it is a Round-Robin model.
6. In a time-sharing system, the user departs from the system as soon as his service requirement is satisfied.
The length of time needed for each pass is $q = r + \theta$ except for the last pass when user leaves as soon as the service is completed. If the remaining service needed is less than θ we denote by "d" the service needed in the last pass.
7. There is a finite supply of customers.
8. The measure of efficiency is the expected response time conditional on the service requirement.
9. An interrupted unit carries with it the knowledge of the processing already done to it. However, this function takes some time, as described above in 3.

Analytical Solution.¹ In order to study the model in steady-state we need to obtain the total time devoted to a customer and the number of quanta already given to a customer.

Denoting the length of a quantum by Q , we have

$$Q \stackrel{d}{=} \begin{cases} r + \theta & \text{with probability } a = e^{-\mu\theta} \\ D & \text{with probability } 1 - a \end{cases}$$

$\stackrel{d}{=}$ denoting equal in distribution, and D having a density $fd(\cdot)$

$$fd(x) = \mu e^{-\mu(x-r)} / (1 - a), \quad r \leq x \leq \theta + r$$

The expectation of the quantum's length is easily derived as

$$E(Q) = r + (1 - a)/\mu$$

The total time devoted by the server to a single customer, including setting-up times, is denoted by S :

$$S = (R - 1)(r + \theta) + D,$$

where

$$E(R) = 1/(1 - a)$$

The procedure to solve this model involves the Markov chains encountered in the machine interference model to study the steady-state number of customers in the system. (The machine interference model has been reviewed in chapter four). It uses the simple argument that the expected number of departures is equal to the expected number of arrivals for a steady state system. The following results, for which a proof is given in appendix thirteen, have been obtained.

The total response time (t) is given as follows

$$E(t) = NE(S)/(1 - P_0) - 1/\lambda$$

when P_0 is the stationary probability of the server being idle. Another important conclusion given is the expected response time given the length of service required².

¹I. Adiri and B. Avi-Itzhak, "A Time-Sharing Queue with a Finite Number of Customers", Journal of ACM. Vol. 16, No. 2, (April 1969), pp. 315-323,

²The proof is given in appendix thirteen. It has been obtained from:
I. Adiri and B. Avi-Itzhak, op. cit.

$$E(T | L=1) = E(T) - 1/\mu - 1 \\ + (r + E(Q)(1 - \beta)(N-1)) \left(\frac{1}{\theta} - \lambda(1-a) \right) \\ + E(Q)\beta \left(\sum_{i=1}^{L-1} E(K_i) - \sum_{i=1}^{\infty} E(K_i) \right)$$

where k is the steady state number of customers, $\beta = e^{-\lambda(\theta+r)}$ and n is the number of customers.

The reader will notice that the procedure (given in appendix thirteen) is different from the one suggested by Coffman and Kleinrock since in this case there are a fixed number of customers. In the previous model there was an infinite number of customers who sent messages to a unique computer. It was assumed that one could consider the arrivals to the computer as a Poisson process. In the present model the n customers are analog to the n machines which break down with intensity equal to λ .

The behavior of the system is difficult to see from the mathematical expressions. Therefore, a graphical representation is used.

Figures 7.1.2 and 7.1.3 show the relation between the expected response time $E(T)$ and the quantum processing size (θ) for a variable number of terminals n where n is a constant. Therefore, one could see the effect of the quantum size on the expected response time. One should recall that there is an overhead factor and that each time a unit is interrupted, a setting-up time r is needed. The quantum size is θ . Figure 7.1.2 shows the relationship between the expected response time $E(t)$ and θ when $N\lambda/\mu = 0.05$. Figure 7.1.3 does the same for $N\lambda/\mu = 1.0$. The reader should recall that $N\lambda/\mu$ is not necessarily the utilization because of the presence of the overhead factor. One can easily see from both figures that as N increases, i.e. as the finite number of users increases so that $N\lambda$ is kept constant, the expected total response time increases. Adiri and Avi-Itzhak used the machine interference

model to solve the model. As one will recall from Chapter Four, this is equivalent to having n machines and a single repairman. As the number of machine N increases so that $N\lambda$ is constant, the value of λ for each user, origin, or source decreases. As N increases there are more available users not in the system who can send messages. If N is very small, e.g. 2, and if the two users are already in the queue there are no more arrivals. If there are 3 users and 2 in the queue, there is still one user who can send requests. If N increases and λ decreases so that $N\lambda$ is a constant there are more independent users who can send requests. As N approaches ∞ , the system becomes^{the} equivalent of a single source sending messages at a rate λ' . As n increases this model approaches the model studied in chapter six and the machine interference model no longer applies.

Figures 7.1.3 to 7.1.7 show the expected total response time as a function of the length of the service required. Different graphs are shown for several values of $N\lambda$ and for several quantum sizes. In the previous chapter we showed on figure 6.1.3 the expected response time as a function of the number of quanta for a Round-Robin with^{an} infinite number of customers which is the same meaning. The number of quanta used by a unit expresses the length of the service required.

Hypothesis.

The expected total response time is properly obtained from the analytical solution.

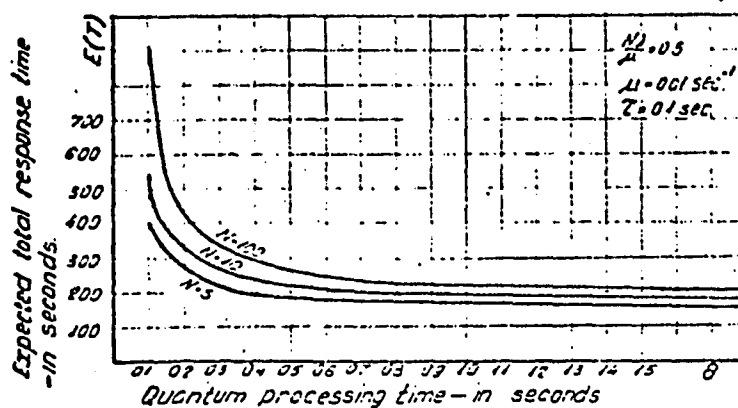


Fig: 7.1.2 Expected total response time as a function of the quantum processing time

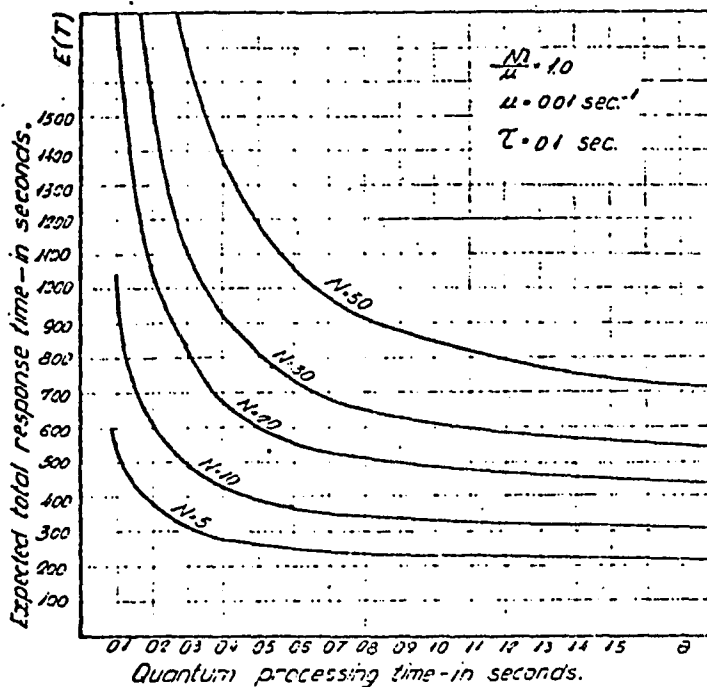


Fig: 7.1.3 Expected total response time as a function of the quantum processing time

SOURCE: I. Adiri and B. Avi-Itzhak, "A Time Sharing Queue with Finite Number of Customers," Journal of ACM, Vol. 10, No. 2, (April 1969), p. 301.

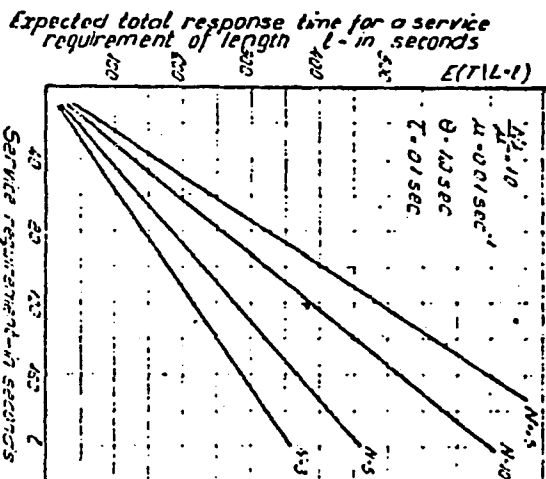


Fig: 7.1.4 Expected total response time for a service requirement of length l as a function of l .

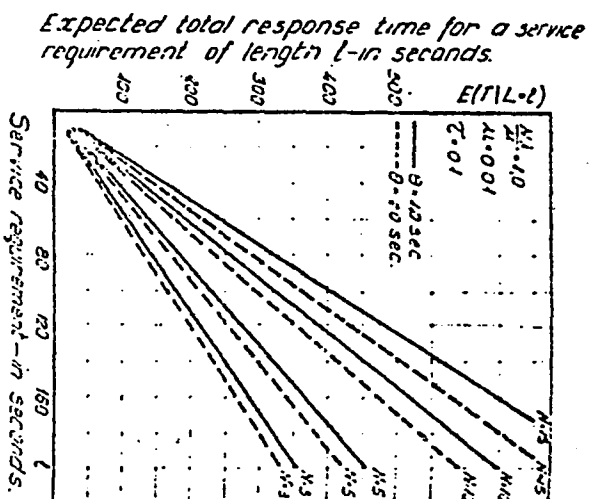


Fig: 7.1.5 Expected total response time for a service requirement of length l as a function of l .

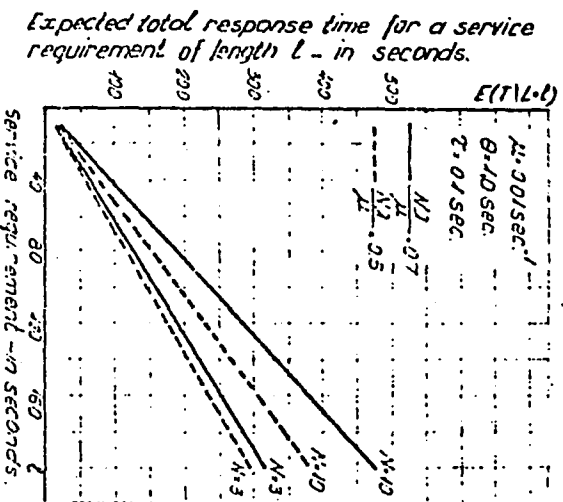


Fig: 7.1.6 Expected total response time for a service requirement of length l as a function of l .

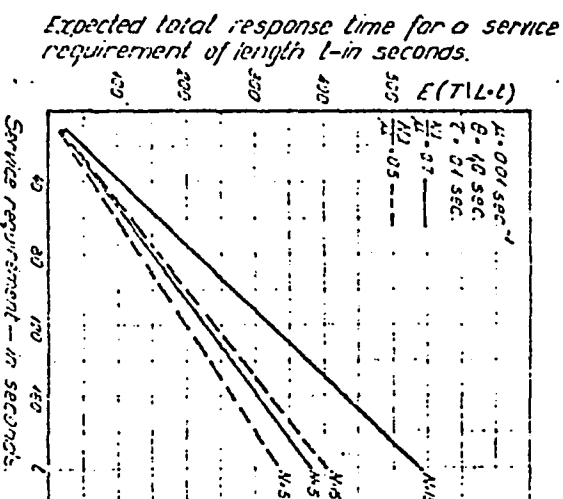


Fig: 7.1.7 Expected total response time for a service requirement of length l as a function of l .

SOURCE: J. Axtel and B. Avi-Itzhak, "A Time Sharing Queue with Finite Number of Customers," JOURNAL OF A.C.S. Vol. 10, No. 2, (April 1969) pp. 328.

Simulation Results. In order to test the hypothesis, the GPSS program TS70 listed in appendix nine was written. The program can be used for many customers, for many queues, or only for one source.

The reader should notice that the examples presented in the previous section were for an average service requirement of 100 seconds and for a quantum size varying from 0.1 second to 10 seconds. This implies that the program must simulate for a quantum of 0.1 second an average of 1000 passes for each group of service requirements.

The program is thus very time consuming; the computer time necessary to run the model is a function of the quantum size. For example, it took 4 hours of computer time to make 8 independent runs of 100 customers for a quantum size of 0.1 second³. The same procedure must be repeated for different quantum sizes and different utilization sizes. For this reason only a few points were selected to be tested.

The expected response times of several simulation runs have been summarized in table 7.2.1. Examples of the actual output of the program are given in tables 7.2.3, 7.2.4, 7.2.5, and 7.2.6. The reader will notice that in addition to the usual output of GPSS discussed earlier, this program has produced the conditional expected waiting time. In tables 7.2.5 and 7.2.6 one can see different tables from Xtim2 to Xtim6. These tables give the distribution of the response

³ These figures are for an IBM 360 model 30 with 64,000 positions of core storage under DOS.

SUMMARY OF THE RESPONSE TIME
ADIRI AND AVI-ITZHAK MODEL
WITH LIMITED NUMBER OF CUSTOMERS

DESCRIPTION	AVERAGE CONDITIONAL RESPONSE TIME DEPENDING ON SERVICE REQUIRED											
			0<S<40		40<S<80		80<S<120		120<S<160		S> 160	
NL/U=0.3 Q=1.0 • N=10 O=0.1 R=1 •	MEANS	289 178	65 37	155 102	225 193	335 262	829 493					
		246 164	45 35	147 123	265 153	473 243	553 407					
		219 263	45 36	117 160	288 229	280 347	586 614					
		410 198	56 46	252 252	407 157	724 210	944 507					
	1/U=100	AVER.	245	45	144	239	359	616				
	CONF.	180 - 311	37 - 54	102 - 185	170 - 300	219 - 499	395 - 837					
	INTER											
NL/U=1.0 Q=1.0 • N=10 O=0.1 1/N=100 R=1 •	MEANS	1275 2404	332 603	962 1748	1452 2768	1917 3763	2479 4614					
		1184 1504	410 520	1071 1524	1517 2104	2051 3215						
	AVER.	1591	466	1326	239	2736	4325					
	CONF.	737-2446	2283-649	756-1895	1059-2895	1359-4113	1368 -7281					
		INTER										
NL/U=1.0 1/U=100 R=1 • N=10 O=0.1 Q=10 •	MEANS	882 1815	253 476	622 1148	780 1934	1182 2605	2089 5052					
		1116 1886	352 376	885 1071	1084 1771	2262 2248	2295 4621					
		668 597	193 140	503 476	890 588	1059 1044	1723 1347					
		2002 462	479 132	1333 313	2171 341	3163 659	4726 1077					
		2551 436	607 85	1650 260	2590 634	3563 664	6526 1023					
	AVER.	924 477	196 96	284 588	998 598	1200 528	2068 1299					
	CONF.	504 1611	171 397	473 1048	734 1661	1160 3077	1895 3785					
	INTER											
NL/U=0.5 1/U=100 R=1 • N=10 O=0.1	MEANS	227 152	53 33	118 87	185 157	262 220	650 422					
		198 146	39 33	110 103	229 135	321 219	451 320					
		178 227	38 31	102 148	230 194	218 303	472 536					
		309 177	48 36	192 85	298 129	536 166	710 456					
		263 207	48 33	143 116	181 302	443 249	685 538					
	169 203	34 42	88 124	187 198	215 268	448 576						
	AVER.	204	39	118	202	289	482					
	CONF.	174 234	34 44	99 137	167 237	219 353	372 592					
	INTER											
NL/U=0.5 O=0.5 N=10 O=0.1 R=1 1/U=100	MEANS	365 217	78 47	216 118	278 214	427 341	1033 583					
		251 265	65 67	150 185	225 256	352 322	986 456					
		352 333	82 79	221 219	233 282	425 435	1001 998					
	AVER.	297	69	184	336	391	842					
		CONF.	233 360	56 82	140 229	257 416	328 454	578 1107				
	INTER.											

NOTE: S = LENGTH OF SERVICE; Q = QUANTUM SIZE IN SECONDS; U = SERVICE RATE;
 1/U = MEAN SERVICE; R = NUMBER OF QUEUES; L = ARRIVAL RATE;
 O = OVERHEAD; CONF. INTER. = 95% CONFIDENCE INTERVALS.
 N = NUMBER OF SOURCES OR THE NUMBER OF USERS

TABLE 7.2.1

SOURCE: SUMMARY OF MANY RUNS OF THE PROGRAM TS70 WRITTEN BY THE AUTHOR
 AND LISTED IN APPENDIX NINE.

UTILIZATION OF THE CPU					
FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN	SEIZING TRANS. NO.	PREFEMPTING TRANS. NO.
CPU	1.000	9756	10.956		
WAITING TIME FOR THE CPU					
QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS
1	1	.000	96	9	9.3
2	1	.000	94	90	95.8
3	1	.000	94	88	93.6
4	1	.000	94	88	93.6
5	1	.001	93	85	91.3
6	1	.001	93	83	89.2
7	1	.001	92	80	86.9
8	1	.002	92	78	84.7
9	23	15.800	8924	0	0
10					
AVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES				189.263	189.263
					10

TIME SPENT IN THE SYSTEM							
TABLE	XTIME	MEAN ARGUMENT		STANDARD DEVIATION		SUM OF ARGUMENTS	
ENTRIES IN TABLE	100	18978.476		17600.000		1897848.000	
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN	
0	0	.00	.0	100.0	.000	-.069	
1000	6	5.99	5.9	94.0	.052	-.917	
2000	6	5.99	11.9	88.0	.105	-.866	
3000	6	5.99	17.9	82.0	.158	-.815	
4000	3	2.99	20.9	79.0	.210	-.764	
5000	2	1.99	22.9	77.0	.263	-.713	
6000	4	3.99	26.9	73.0	.316	-.662	
7000	4	3.99	30.9	69.0	.369	-.611	
8000	6	5.99	36.9	63.0	.421	-.560	
9000	7	6.99	43.9	56.0	.474	-.509	
10000	2	1.99	45.9	54.0	.526	-.458	
11000	3	2.99	48.9	51.0	.579	-.407	
12000	4	3.99	52.9	47.0	.632	-.356	
13000	3	2.99	55.9	44.0	.684	-.305	
14000	1	.99	56.9	43.0	.737	-.254	
15000	1	.99	57.9	42.0	.790	-.202	
16000	0	.00	57.9	42.0	.843	-.151	
17000	0	.00	57.9	42.0	.895	-.100	
18000	1	.99	58.9	41.0	.948	-.049	
19000	2	1.99	60.9	39.0	1.001	.001	
20000	1	.99	61.9	38.0	1.053	.052	
21000	3	2.99	64.9	35.0	1.106	.103	
22000	3	2.99	67.9	32.0	1.159	.154	
23000	1	.99	68.9	31.0	1.211	.205	
24000	3	2.99	71.9	28.0	1.264	.256	
25000	3	2.99	74.9	25.0	1.317	.307	
26000	1	.99	75.9	24.0	1.369	.358	
27000	1	.99	76.9	23.0	1.422	.409	
28000	1	.99	77.9	22.0	1.475	.460	
29000	0	.00	77.9	22.0	1.528	.511	
30000	2	1.99	79.9	20.0	1.580	.562	
31000	0	.00	79.9	20.0	1.633	.613	
32000	1	.99	80.9	19.0	1.686	.664	
33000	1	.99	81.9	18.0	1.739	.715	
34000	0	.00	81.9	18.0	1.791	.766	
35000	1	.99	82.9	17.0	1.844	.817	
36000	1	.99	83.9	16.0	1.896	.868	
37000	1	.99	84.9	15.0	1.949	.919	
38000	1	.99	85.9	14.0	2.002	.970	
39000	0	.00	85.9	14.0	2.054	1.021	
40000	1	.99	86.9	13.0	2.107	1.072	
41000	1	.99	86.9	13.0	2.159	1.123	

Table 7.2.3

Sample output of the program TS70 listed in appendix nine

TABLE PASS ENTRIES IN TABLE		MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS	NON-WEIGHTED	
100		104.299	99.187	10430.000		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-1.051
1	2	1.99	1.9	98.0	.009	-1.041
2	0	.00	1.9	98.0	.019	-1.031
3	0	.00	1.9	98.0	.028	-1.021
4	0	.00	1.9	98.0	.038	-1.011
5	1	.99	2.9	97.0	.047	-1.001
6	0	.00	2.9	97.0	.057	-.991
7	1	.99	3.9	96.0	.067	-.980
8	0	.00	3.9	96.0	.076	-.970
9	0	.00	3.9	96.0	.086	-.960
10	2	1.99	5.9	94.0	.095	-.950
11	0	.00	5.9	94.0	.105	-.940
12	0	.00	5.9	94.0	.115	-.930
13	1	.99	6.9	93.0	.124	-.920
14	0	.00	6.9	93.0	.134	-.910
15	1	.99	7.9	92.0	.143	-.900
16	3	2.99	10.9	89.0	.153	-.890
17	0	.00	10.9	89.0	.162	-.880
18	1	.99	11.9	88.0	.172	-.870
19	1	.99	12.9	87.0	.182	-.859
20	1	.99	13.9	86.0	.191	-.849
21	1	.99	14.9	85.0	.201	-.839
22	3	2.99	17.9	82.0	.210	-.829
23	0	.00	17.9	82.0	.220	-.819
24	0	.00	17.9	82.0	.230	-.809
25	2	1.99	19.9	80.0	.239	-.799
26	0	.00	19.9	80.0	.249	-.789
27	0	.00	19.9	80.0	.259	-.779
28	0	.00	19.9	80.0	.268	-.769
29	0	.00	19.9	80.0	.278	-.759
30	1	.99	20.9	79.0	.287	-.749
31	0	.00	20.9	79.0	.297	-.739
32	0	.00	20.9	79.0	.306	-.728
33	0	.00	20.9	79.0	.316	-.718
34	2	1.99	22.9	77.0	.325	-.709
35	0	.00	22.9	77.0	.335	-.698
36	2	1.99	24.9	75.0	.345	-.688
37	1	.99	25.9	74.0	.354	-.678
38	2	1.99	27.9	72.0	.364	-.668
39	0	.00	27.9	72.0	.373	-.658
40	0	.00	27.9	72.0	.383	-.648
41	1	.99	28.9	71.0	.393	-.638
42	1	.99	29.9	70.0	.402	-.628
43	0	.00	29.9	70.0	.412	-.618
44	0	.00	29.9	70.0	.421	-.607
45	2	1.99	31.9	68.0	.431	-.597
46	4	3.99	35.9	64.0	.441	-.587
47	0	.00	35.9	64.0	.450	-.577
48	0	.00	35.9	64.0	.460	-.567
49	1	.99	36.9	61.0	.469	-.557
50	3	2.99	39.9	60.0	.479	-.547
51	1	.99	40.9	59.0	.488	-.537
52	0	.00	40.9	59.0	.498	-.527
53	0	.00	40.9	59.0	.508	-.517
54	1	.99	41.9	58.0	.517	-.507
55	0	.00	41.9	58.0	.527	-.497
56	0	.00	41.9	58.0	.536	-.486
57	0	.00	41.9	58.0	.546	-.476
58	1	.99	42.9	57.0	.556	-.466
OVERFLOW	57	56.99	100.0	.0		
AVERAGE VALUE OF OVERFLOW		100.00				

Table 7.2.4

Sample output of the program TS70 listed in appendix nine

TABLE XY1M2 ENTRIES IN TABLE		MEAN ARGUMENT 2829.606	STANDARD DEVIATION 2303.000	SUM OF ARGUMENTS 79117.000	NON-WEIGHTED	
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-1.226
1000	6	21.42	21.4	78.5	.353	-.792
2000	6	21.42	42.8	57.1	.707	-.358
3000	6	21.42	64.2	35.7	1.061	.075
4000	3	10.71	74.9	25.0	1.415	.509
5000	2	7.14	82.1	17.8	1.769	.944
6000	1	3.57	85.7	14.2	2.123	1.378
7000	2	7.14	92.8	7.1	2.477	1.812
8000	1	3.57	96.4	3.5	2.831	2.246
9000	1	3.57	100.0	.0	3.185	2.681

REMAINING FREQUENCIES ARE ALL ZERO

TABLE XY1M3 ENTRIES IN TABLE		MEAN ARGUMENT 9294.960	STANDARD DEVIATION 2916.000	SUM OF ARGUMENTS 241669.000	NON-WEIGHTED	
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-3.187
1000	0	.00	.0	100.0	.197	-2.844
2000	0	.00	.0	100.0	.395	-2.501
3000	0	.00	.0	100.0	.592	-2.158
4000	0	.00	.0	100.0	.790	-1.815
5000	0	.00	.0	100.0	.987	-1.472
6000	3	11.53	11.5	88.4	1.185	-1.129
7000	2	7.69	19.2	80.7	.753	-.787
8000	5	19.23	38.4	61.5	.960	-.444
9000	9	19.23	57.6	42.3	.960	-.101
10000	2	7.69	65.3	34.6	1.075	.241
11000	3	11.53	76.9	23.0	1.183	.584
12000	1	3.84	80.7	19.2	1.291	.927
13000	2	7.69	88.4	11.5	1.398	1.270
14000	1	3.84	92.3	7.6	1.506	1.613
15000	1	3.84	96.1	3.8	1.613	1.956
16000	0	.00	96.1	3.8	1.721	2.299
17000	0	.00	96.1	3.8	1.829	2.642
18000	1	3.84	100.0	.0	1.936	2.985

REMAINING FREQUENCIES ARE ALL ZERO

TABLE XY1M4 ENTRIES IN TABLE		MEAN ARGUMENT 17919.664	STANDARD DEVIATION 5216.000	SUM OF ARGUMENTS 268795.000	NON-WEIGHTED	
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-3.435
1000	0	.00	.0	100.0	.055	-3.243
2000	0	.00	.0	100.0	.111	-3.052
3000	0	.00	.0	100.0	.167	-2.860
4000	0	.00	.0	100.0	.223	-2.668
5000	0	.00	.0	100.0	.279	-2.476
6000	0	.00	.0	100.0	.334	-2.285
7000	0	.00	.0	100.0	.390	-2.093
8000	0	.00	.0	100.0	.446	-1.901
9000	1	6.66	6.6	93.3	.502	-1.710
10000	0	.00	6.6	93.3	.558	-1.518
11000	0	.00	6.6	93.3	.613	-1.326
12000	3	19.99	26.6	73.3	.669	-1.134
13000	1	6.66	33.3	66.6	.725	-.943
14000	0	.00	33.3	66.6	.781	-.751
15000	0	.00	33.3	66.6	.837	-.559
16000	0	.00	33.3	66.6	.892	-.368
17000	0	.00	33.3	66.6	.948	-.176
18000	0	.00	33.3	66.6	1.004	.015
19000	2	13.33	46.6	53.3	1.060	.207
20000	1	6.66	53.3	46.6	1.116	.398
21000	2	13.33	66.6	33.3	1.171	.590
22000	2	13.33	79.9	20.0	1.227	.782
23000	1	6.66	86.6	13.3	1.283	.973
24000	0	.00	86.6	13.3	1.339	1.165
25000	2	13.33	100.0	.0	1.395	1.357

REMAINING FREQUENCIES ARE ALL ZERO

Table 7.2.5

Sample output of the program MS20 listed in appendix nine

TABLE XTIME ENTRIES IN TABLE		MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS		NON-WEIGHTED
14		27234.710	5296.000	381286.000		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-5.142
1000	0	.00	.0	100.0	-.036	-4.953
2000	0	.00	.0	100.0	-.073	-4.764
3000	0	.00	.0	100.0	-.110	-4.576
4000	0	.00	.0	100.0	-.146	-4.387
5000	0	.00	.0	100.0	-.183	-4.198
6000	0	.00	.0	100.0	-.220	-4.009
7000	0	.00	.0	100.0	-.257	-3.820
8000	0	.00	.0	100.0	-.293	-3.631
9000	0	.00	.0	100.0	-.330	-3.443
10000	0	.00	.0	100.0	-.367	-3.254
11000	0	.00	.0	100.0	-.403	-3.065
12000	0	.00	.0	100.0	-.440	-2.876
13000	0	.00	.0	100.0	-.477	-2.687
14000	0	.00	.0	100.0	-.514	-2.499
15000	0	.00	.0	100.0	-.550	-2.310
16000	0	.00	.0	100.0	-.587	-2.121
17000	0	.00	.0	100.0	-.624	-1.932
18000	0	.00	.0	100.0	-.660	-1.743
19000	0	.00	.0	100.0	-.697	-1.554
20000	0	.00	.0	100.0	-.734	-1.366
21000	1	7.14	7.1	92.8	-.771	-1.177
22000	1	7.14	14.2	85.7	-.807	-.988
23000	0	.00	14.2	85.7	-.844	-.799
24000	3	21.42	35.7	64.2	-.881	-.610
25000	1	7.14	42.8	57.1	-.917	-.421
26000	1	7.14	49.9	50.0	-.954	-.233
27000	1	7.14	57.1	42.8	-.991	-.044
28000	1	7.14	64.2	35.7	1.028	.144
29000	0	.00	64.2	35.7	1.064	.333
30000	1	7.14	71.4	28.5	1.101	.522
31000	0	.00	71.4	28.5	1.138	.710
32000	1	7.14	78.5	21.4	1.174	.899
33000	1	7.14	85.7	14.2	1.211	1.088
34000	0	.00	85.7	14.2	1.248	1.277
35000	0	.00	85.7	14.2	1.285	1.466
36000	1	7.14	92.8	7.1	1.321	1.655
37000	0	.00	92.8	7.1	1.358	1.843
38000	1	7.14	100.0	.0	1.395	2.032

REMAINING FREQUENCIES ARE ALL ZERO

TABLE XTIME ENTRIES IN TABLE		MEAN ARGUMENT	STANDARD DEVIATION	SUM OF ARGUMENTS		NON-WEIGHTED
17		54528.292	17264.000	926981.000		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-3.158
1000	0	.00	.0	100.0	-.019	-3.100
2000	0	.00	.0	100.0	-.036	-3.042
3000	0	.00	.0	100.0	-.055	-2.984
4000	0	.00	.0	100.0	-.073	-2.926
5000	0	.00	.0	100.0	-.091	-2.868
6000	0	.00	.0	100.0	-.110	-2.810
7000	0	.00	.0	100.0	-.129	-2.753
8000	0	.00	.0	100.0	-.146	-2.695
9000	0	.00	.0	100.0	-.165	-2.637
10000	0	.00	.0	100.0	-.183	-2.579
11000	0	.00	.0	100.0	-.201	-2.521
12000	0	.00	.0	100.0	-.220	-2.463
13000	0	.00	.0	100.0	-.238	-2.405
14000	0	.00	.0	100.0	-.256	-2.347
15000	0	.00	.0	100.0	-.275	-2.289
16000	0	.00	.0	100.0	-.293	-2.231
17000	0	.00	.0	100.0	-.311	-2.173
18000	0	.00	.0	100.0	-.330	-2.115
19000	0	.00	.0	100.0	-.348	-2.057
20000	0	.00	.0	100.0	-.366	-2.000
21000	0	.00	.0	100.0	-.385	-1.942
22000	0	.00	.0	100.0	-.403	-1.884
23000	0	.00	.0	100.0	-.421	-1.826
24000	0	.00	.0	100.0	-.440	-1.768
25000	0	.00	.0	100.0	-.458	-1.710
26000	0	.00	.0	100.0	-.476	-1.652
27000	0	.00	.0	100.0	-.495	-1.594
28000	0	.00	.0	100.0	-.513	-1.536
29000	0	.00	.0	100.0	-.531	-1.478

Table 7.2.6
Sample output of the program TS70 listed in appendix nine

time for units requiring between 0 to 40 seconds up to units requiring over 160 seconds; the average requirement is 100 seconds. For each run, the mean response time for each group has been entered in table 7.2.1. Several points have been chosen for analysis as described in the first column of the table 7.2.1. For each point, the table shows the mean response time for each of the 12 runs, the mean response time conditional on the service required for values between 0 and 40, 40 and 80, 80 and 120, 120 and 160, and for values over 160 seconds. An average of the 12 mean response times conditional on the service requirement is given for each of the intervals. A 95% confidence interval is given.

Figure 7.2.1 shows the response time as a function of the quantum size. The results agree with the analytical results presented in figure 7.1.1, i.e. the analytical results are within the 95% confidence interval. However, the simulation runs have been made only for a few quantum sizes and for only 10 customers on the system.

The conditional response time in terms of the service requirement is also illustrated. As one can see by comparing figure 7.1.3 (analytical results) and figure 7.2.2 (simulation results) the simulation response time for $n \lambda/\mu = 1.0$ is significantly larger than the one suggested by the analytical results. The reader should notice that $n \lambda/\mu$ is not the CPU utilization since at each pass one must add the overhead to the quantum size. Figure 7.2.3 illustrates the same relationship for a quantum size of 10 seconds instead of 1.0 second as in the previous illustration.

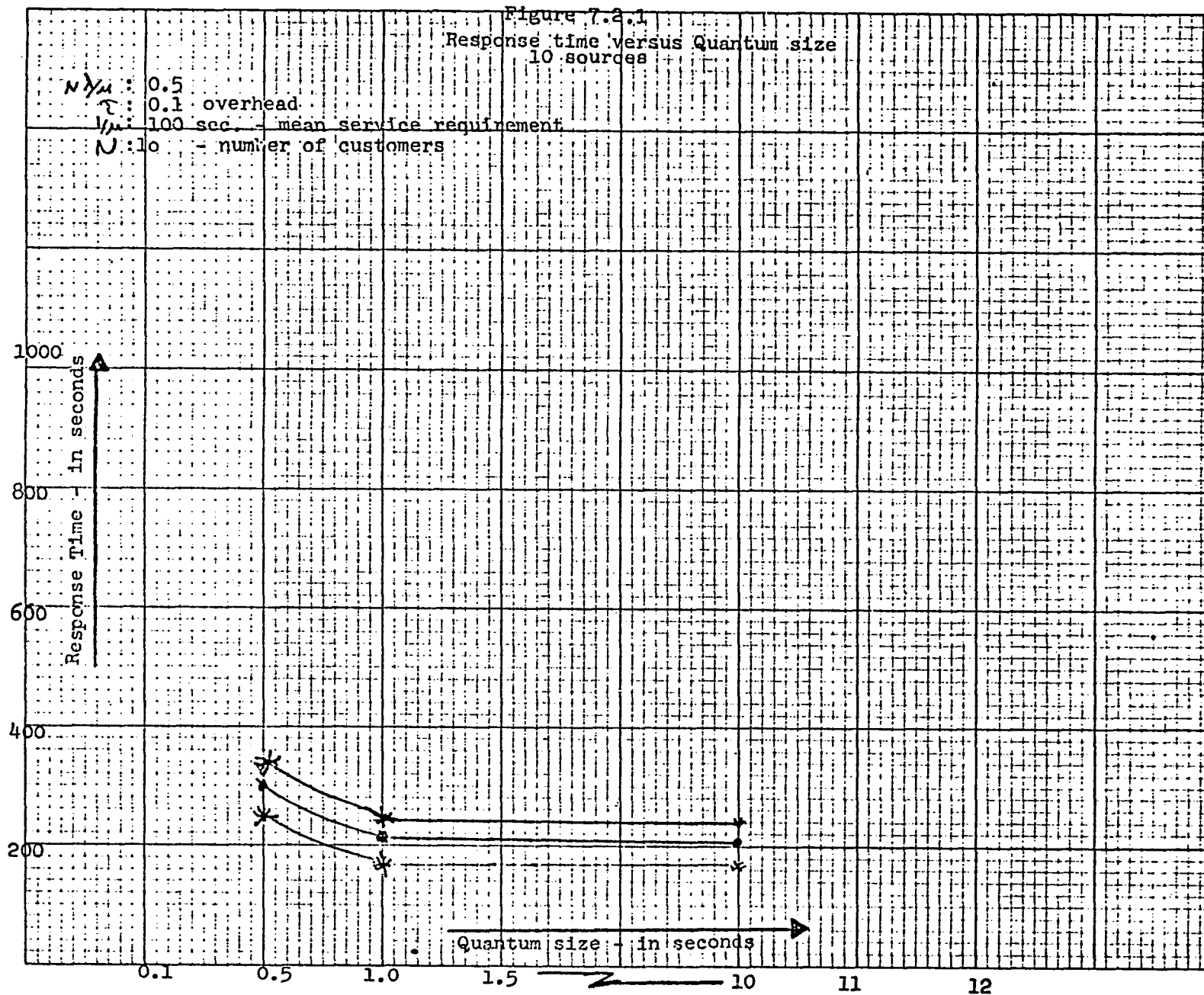
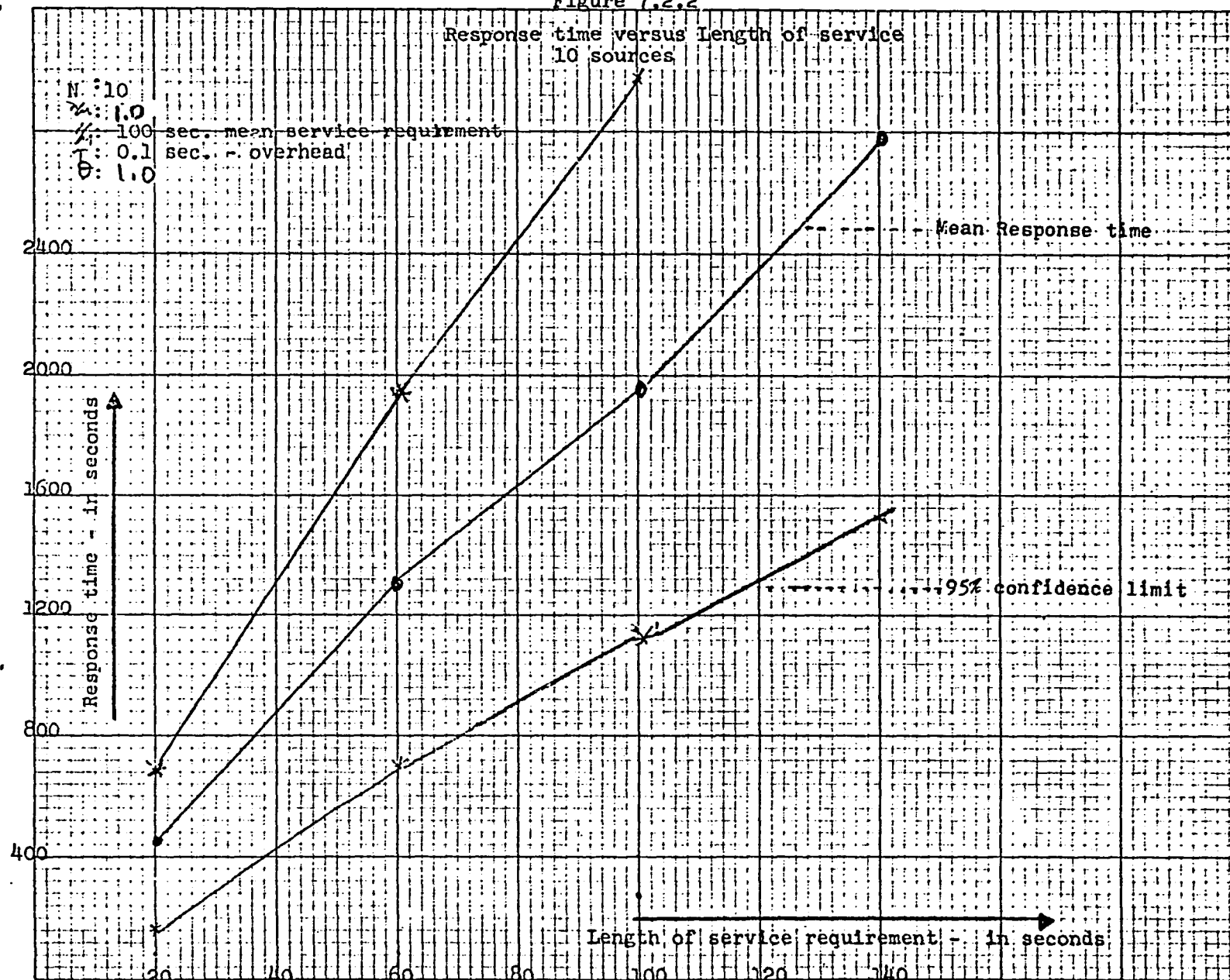
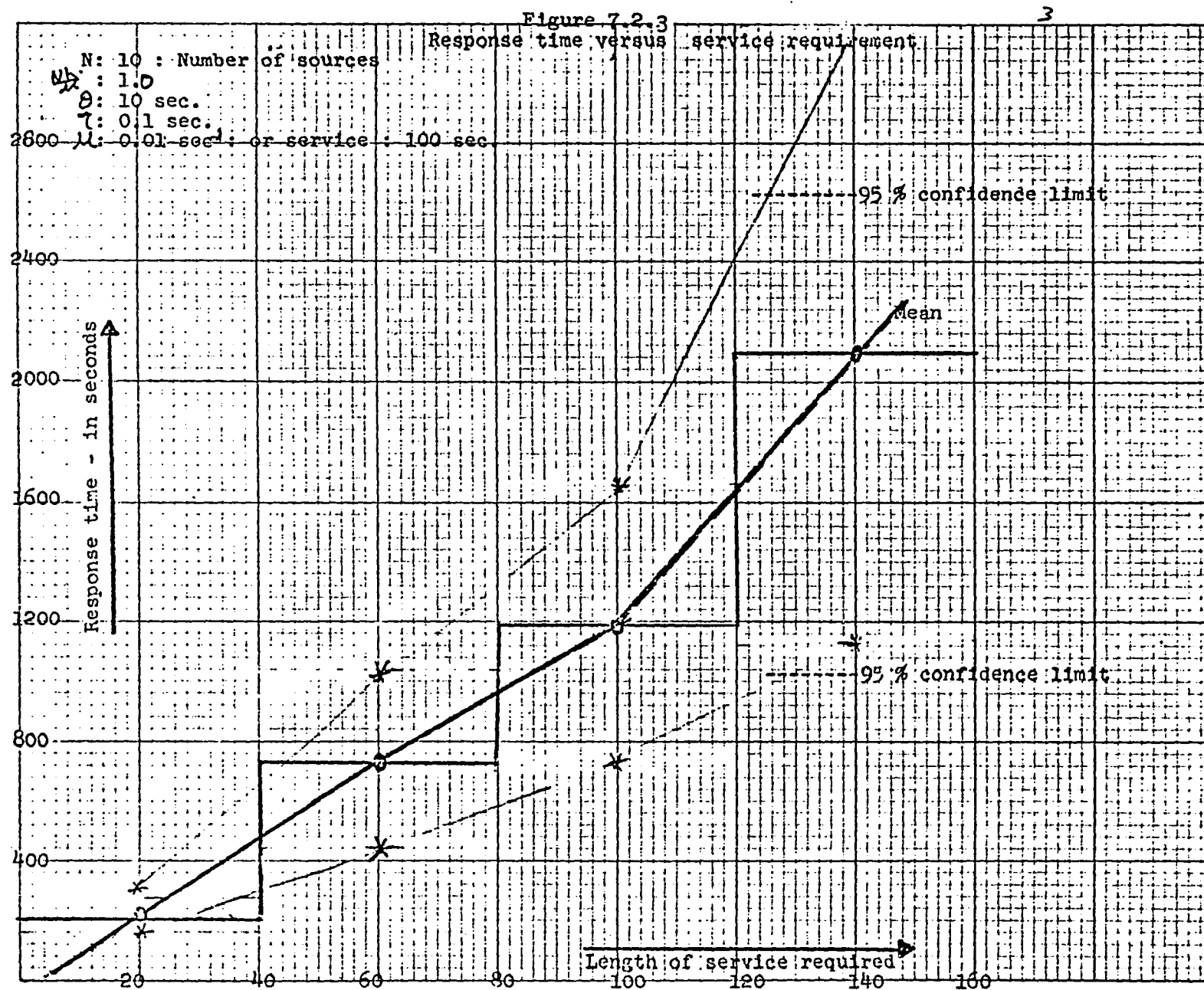


Figure 7.2.2





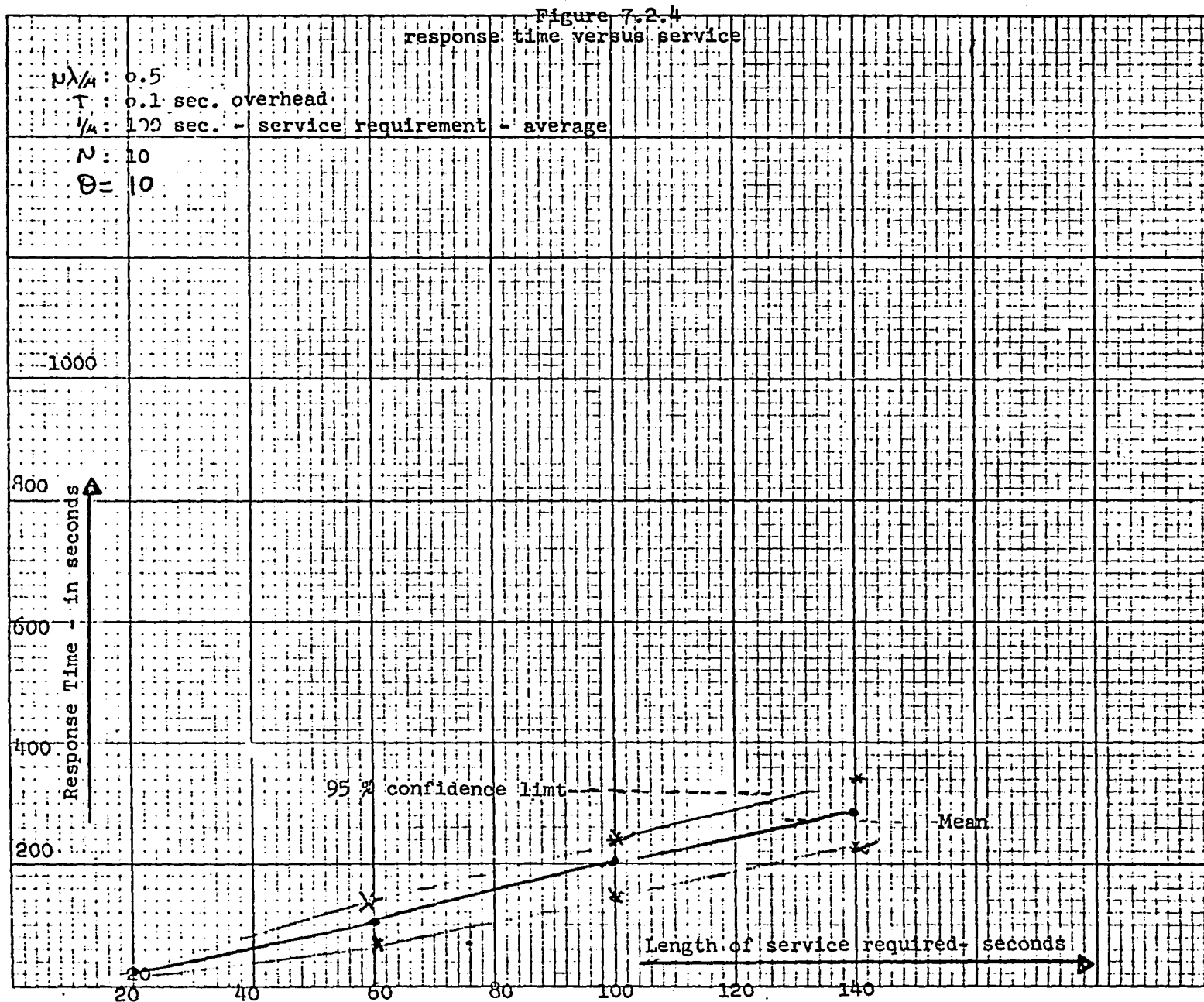
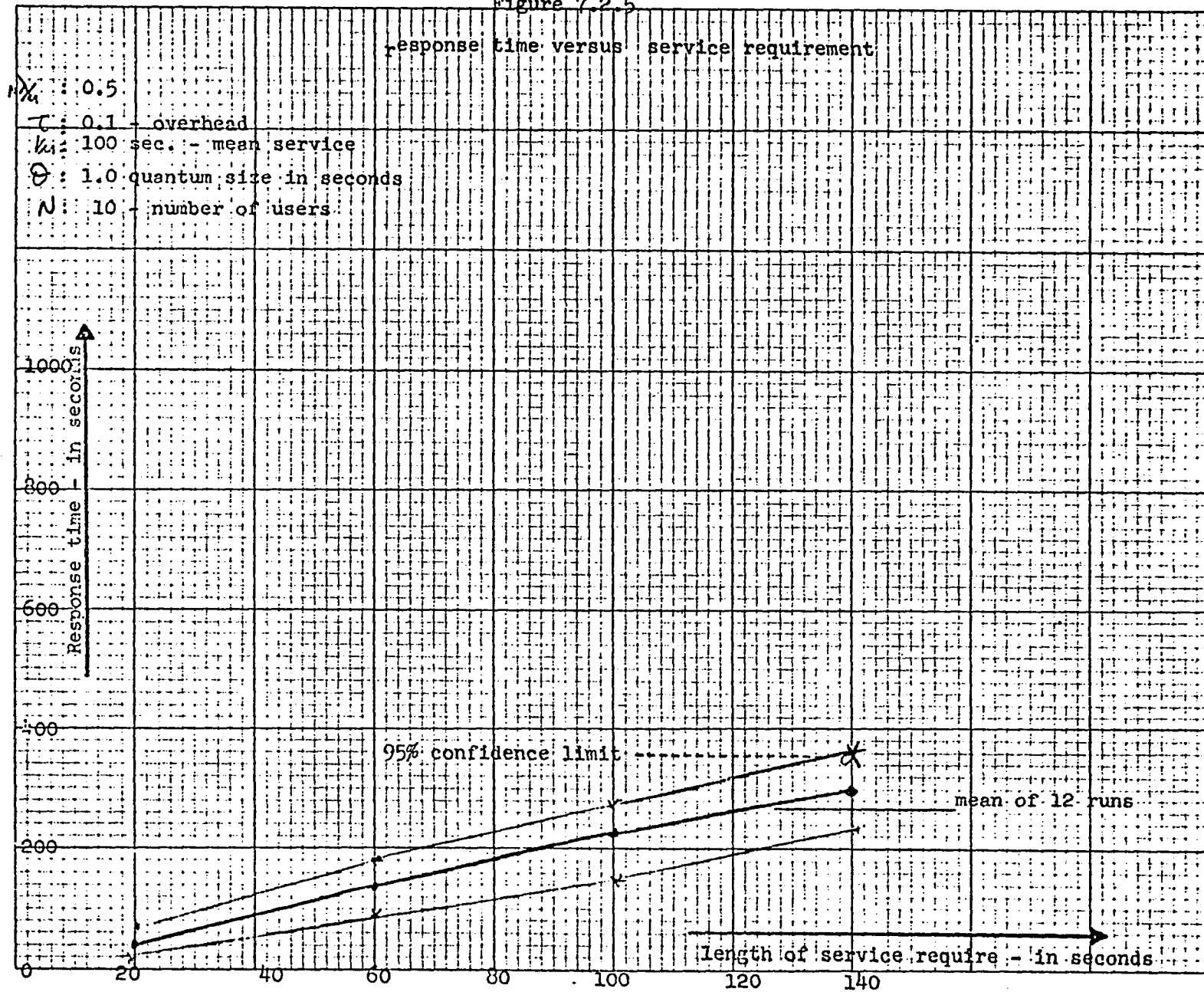


Figure 7.2.5



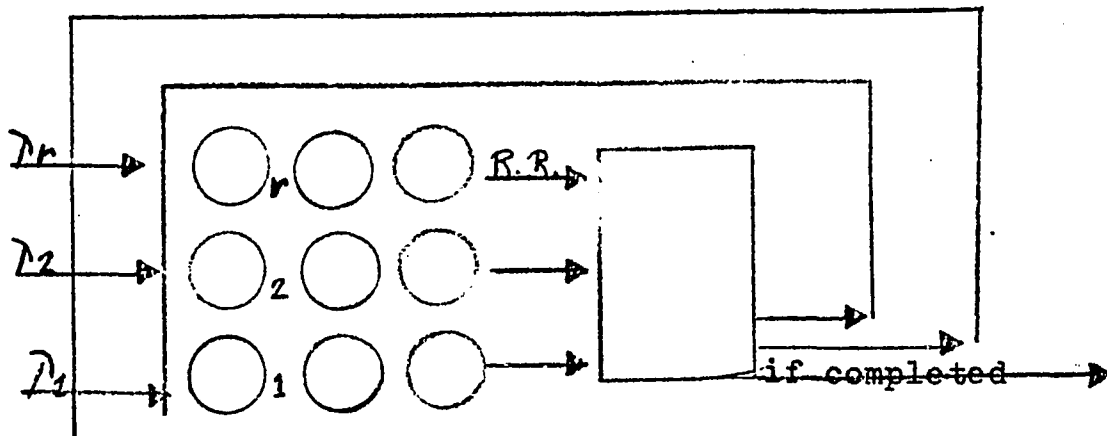
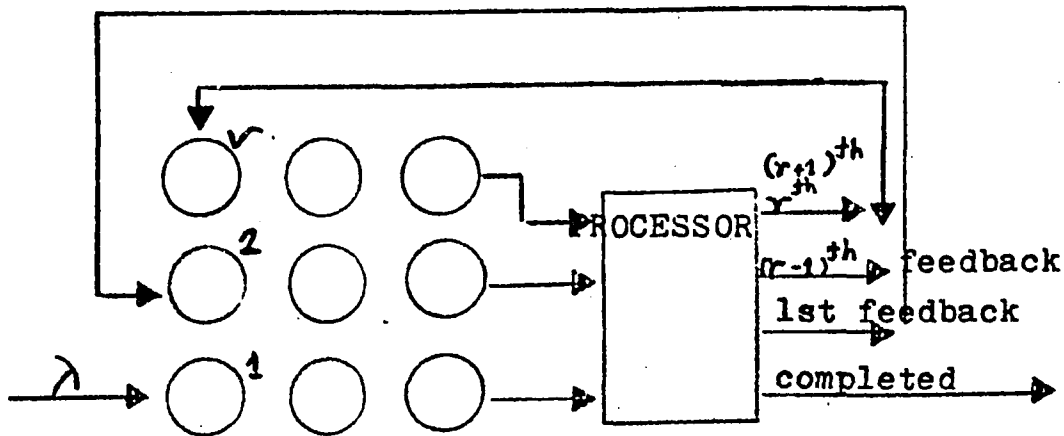


FIGURE 7.3.1. The FB model with r levels of Queues
(upper without priority; lower with priority)

Source: Modification of the figures presented in:

E. Coffman and L. Kleinrock, "Feedback Queuing Model for Time Shared Systems", Journal of ACM, Vol. 15, No. 4, (October 1968), p. 554.

The results of the analytical results are within the confidence limits for $n\lambda/\mu = 0.5$ as illustrated by comparing figure 7.2.4 and figure 7.1.5.

We fail to reject the hypothesis concerning the solution for the steady-state system where ρ , the utilization value, is less than 1. However, Adiri and Avi-Itzhak have extended the same results to an over-loaded system where $n\lambda/\mu = 1.0$, i.e. for a system where $\rho > 1$, since at each pass one must add the overhead value of 0.1. Only in this case do the simulation results differ from the analytical results. We fail to reject the hypothesis that the expected response time for a steady-state system is properly obtained by analytical techniques.

FEEDBACK MODEL WITH LIMITED QUEUES.

In the previous section and in the previous chapter we reviewed a model for a Round-Robin, where the interrupted unit comes back in the same queue, and for the Feedback where the unit goes to a lower queue after each pass. Adiri and Avi-Itzhak have proposed a model called RRn which is like the FB model studied in chapter six but where there are only r levels. If the unit needs more than n passes it always comes back on the r^{th} queue, as shown on figure 7.3.1. In the r^{th} queue, the discipline becomes a Round-Robin. Furthermore there is a certain time lost for swapping, i.e. for interrupting a unit in process.

Parameters and Assumptions. The following assumptions are made.

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

where the mean inter-arrival time is $1/\lambda$ seconds and the mean arrival rate is λ , units per second.

2. For the service process an exponential service distribution is assumed,

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases}$$

where the mean service requirement is $1/\mu$ seconds.

3. The quantum size is a constant value θ .
4. A fixed overhead is added at the end of each quantum r .
5. There is a finite number of queue.
6. In a time-sharing system, the user departs from the system as soon as his service requirement is satisfied. The length of time needed for each pass is $q = r + \theta$ except for the last pass when the user leaves as soon as the service is completed. If the remaining service needed is less than θ , we denote by "d" the service needed in the last pass.

7. All units are assumed to come from a unique source.
8. The measure of efficiency is the expected response time conditional on the service requirement.
9. An interrupted unit carries with it the knowledge of the processing already done to it. However, this function takes some time as described above in 3.
10. There is a finite number of feedback loops r . The r^{th} queue is processed in a Round-Robin manner.

Analytical Solution. The technique used here is to study the behavior of the first $(r - 1)$ queues which is the same as in the FB model and to add to it the behavior of the r^{th} queue which becomes similar to a Round-Robin. The reader will remember that in the FB model the unit goes to a lower level queue after each quantum. In the FB model the unit does the same except that since there are only r queues it will almost come back in the r^{th} queue if it needs more than r passes. The following results have been obtained⁴.

⁴ The proofs of the theorems are given in appendix fourteen. There are from:
I. Adiri and B. Avi-Itzhak, "A Time-Sharing with Many Queues". Operations Research, Vol. 17, No. 6, (November-December 1969). pp. 1077-1089.

. Appendix fourteen also describes the meaning of all the symbols used.

The expected total response time is:

$$E(T) = \int_0^{\infty} E(T|L=l) \mu e^{-\mu l} dl$$

$$= \frac{1}{\mu} + \sum_{n=1}^{\infty} \beta_{n-1} (1 - \alpha_n) \left\{ \frac{\lambda E(U_n^2) + \lambda \sum_{i=n+1}^{\infty} \beta_{i-1} E(Q_i^2)}{2[1 - \lambda E(U_{n-1})][1 - \lambda E(U_n)]} + \frac{q_{n-1}}{1 - \lambda E(U_{n-1})} - \sum_{i=1}^{n-1} \theta_i + \tau_n \right\}.$$

In this expression $E(T|L=l)$ is the total response-time expectation of a customer who demands l service-time units in the RR ∞ model, $\alpha_0 = \beta_0 = 1$, $q_0 = \theta_0 = \tau_0 = 0$, and

$$U_n = \begin{cases} q_{i-1} + D_i & \text{with probability } \beta_{i-1}(1 - \alpha_i), \quad (i=1, 2, \dots, n) \\ q_n & \text{with probability } \beta_n, \quad (n=1, 2, \dots) \end{cases}$$

$$E(U_n) = q_n \beta_n + \sum_{i=1}^n \beta_{i-1} (1 - \alpha_i) [q_{i-1} + E(D_i)],$$

$$E(U_n^2) = q_n^2 \beta_n + \sum_{i=1}^n \beta_{i-1} (1 - \alpha_i) [q_{i-1}^2 + 2q_{i-1} E(D_i) + E(D_i^2)].$$

($n=1, 2, \dots$)

$$\alpha_i = e^{-\mu \theta_i},$$

$$\beta_i = \exp(-\mu \sum_{j=1}^i \theta_j) = \prod_{j=1}^i \alpha_j,$$

$$q_i = \sum_{j=1}^i (\theta_j + \tau_j), \quad (i=1, 2, \dots)$$

The conditional time depending on the length of service required is expressed as:

$$E[T(l)] = [\lambda E(U_{r-1}^2) + \lambda E(Q_r^2) \beta_{r-1} / (1 - \alpha_r)] / 2[1 - \lambda E(U_{r-1})]^2$$

$$+ q_{r-1} / [1 - \lambda E(U_{r-1})]$$

$$+ [(b - \alpha_r) / \lambda \beta_{r-1}] \{ [\alpha_r a + (1 - \alpha_r) E(K_r)] / (1 - \alpha_r b) + [E(K_r) - a / (1 - b)] (1 - b^{n-r}) / (1 - b) + (n - r) a / (1 - b) \} + (n - r) a / (\lambda \beta_{r-1})$$

$$+ l - \sum_{i=1}^{n-1} \theta_i + \tau_r.$$

$E(K_r)$ is obtainable from

$$E(T) = \int_0^{\infty} E[T(l)] \mu e^{-\mu l} dl,$$

where:

$K_i + 1$ is the number of customers present in the i^{th} queue at the time of the i^{th} admission to service.

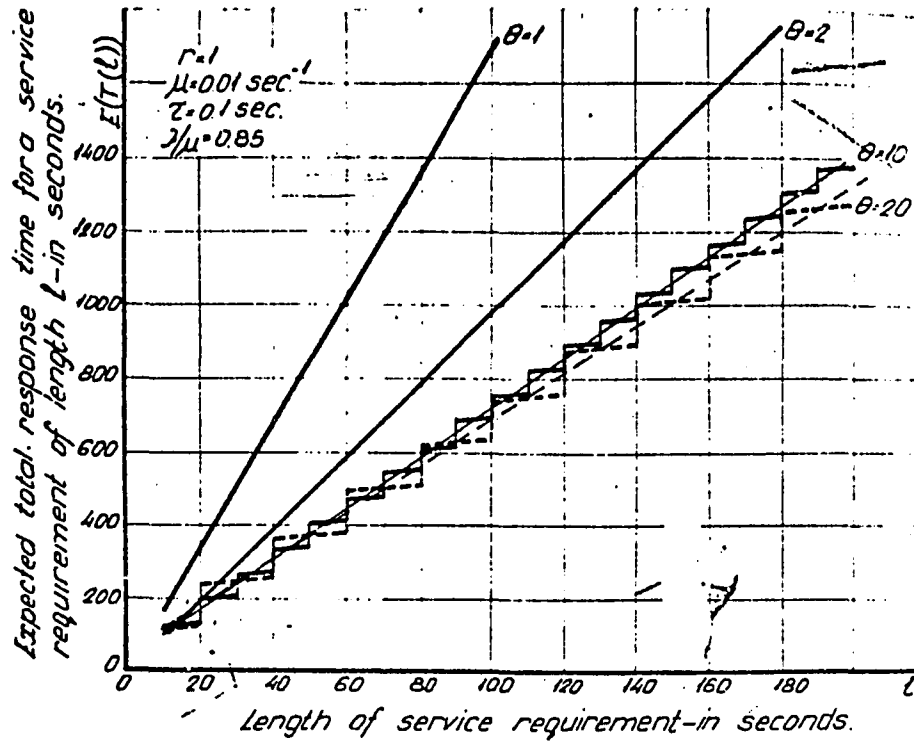
7.3.1

7.3.2

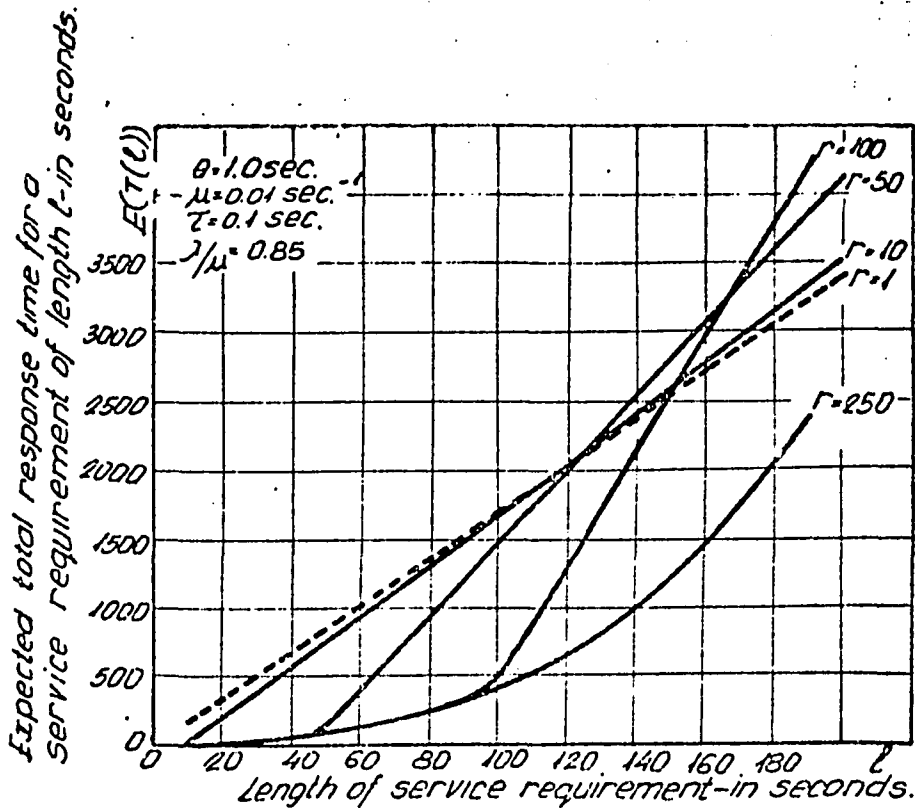
In order to see the behavior of the response time as a function of other variables, some graphical representations are given. Figure 7.3.2 shows the expected response time for the usual Round-Robin model as described in chapter six. As shown in chapter six on figure 6.1.3, and expressed here, the units with short requirements do better, but they still have to wait behind limits of longer requirement. By

increasing r , i.e. the number of feedbacks before we get to the Round-Robin scheduling, we decrease the response time for units needing only a few quanta as shown on figure 7.3.3. As the number of feedbacks increases (before Round-Robin) the advantage to small jobs becomes more obvious as shown on figures 7.3.3 and 7.3.4. In chapter six, we showed the same relationship in terms of the number of quanta to which a unit will have to go in an infinite number of levels model. Referring to figure 7.3.3 one can see that if r is 100, i.e. if there are 99 feedback loops to lower priority queues followed by one Round-Robin queue, the number of units actually in queue 100 is very small but they wait a long time. Figure 7.3.6 shows the effect of increasing the size of the quantum when $\tau/\mu = 0.5$. One can see that the effect is not as obvious as the one shown on figure 7.3.5. The set-up time accounts for the difference.

Figure 7.3.7 shows the effect of saturation on the system. In an overloaded system the unit requiring only a short processing is serviced. This means that at peak hours the units requiring long service may never be serviced until the demand in the system decreases. However, short demands, for example units requiring one quantum, will be serviced even during overloading conditions. This expresses well the

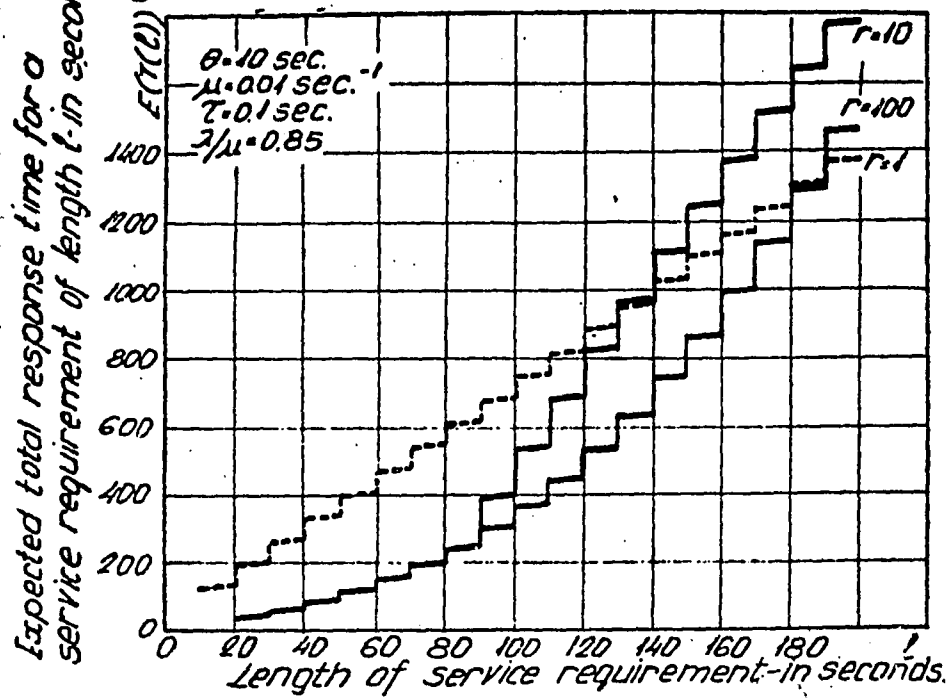


7.3.2 Expected total response time for a service requirement of length l as a function of l (for different values of θ and the RR 1 model).



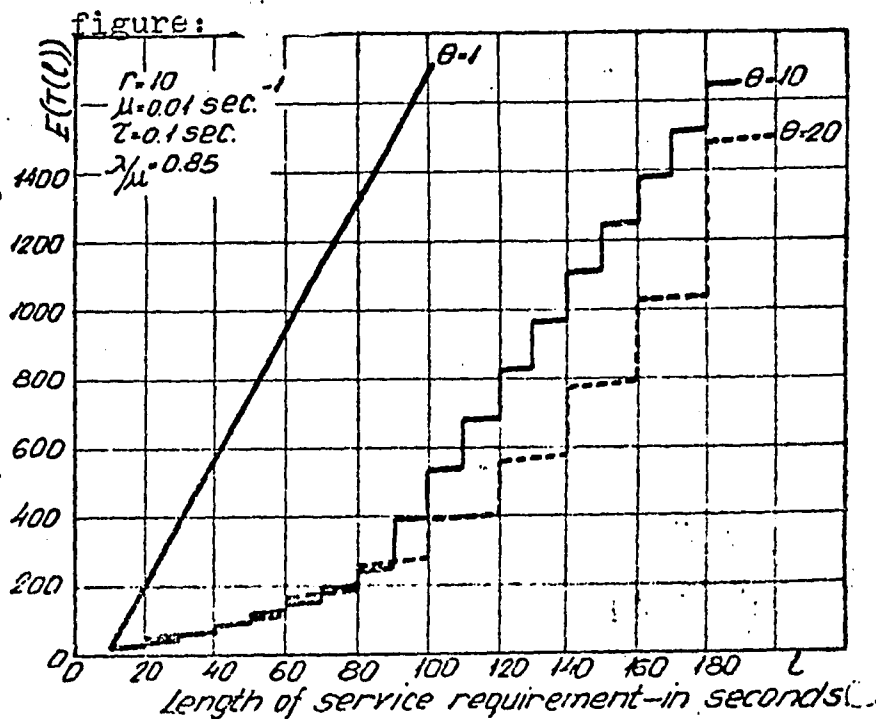
7.3.3 Expected total response time for a service requirement of length l as a function of l (for different values of r).

figure:



7.3.4 Expected total response time for a service requirement of length l as a function of l (for different values of r).

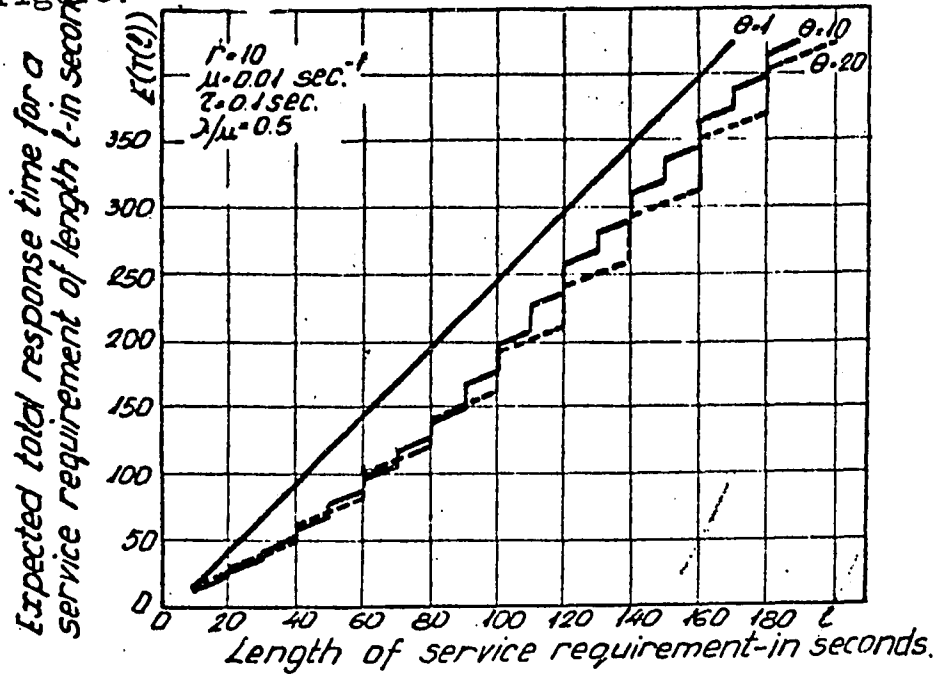
Expected total response time for a service requirement of length l in seconds.



7.3.5 Expected total response time for a service requirement of length l as a function of l (for different values of θ).

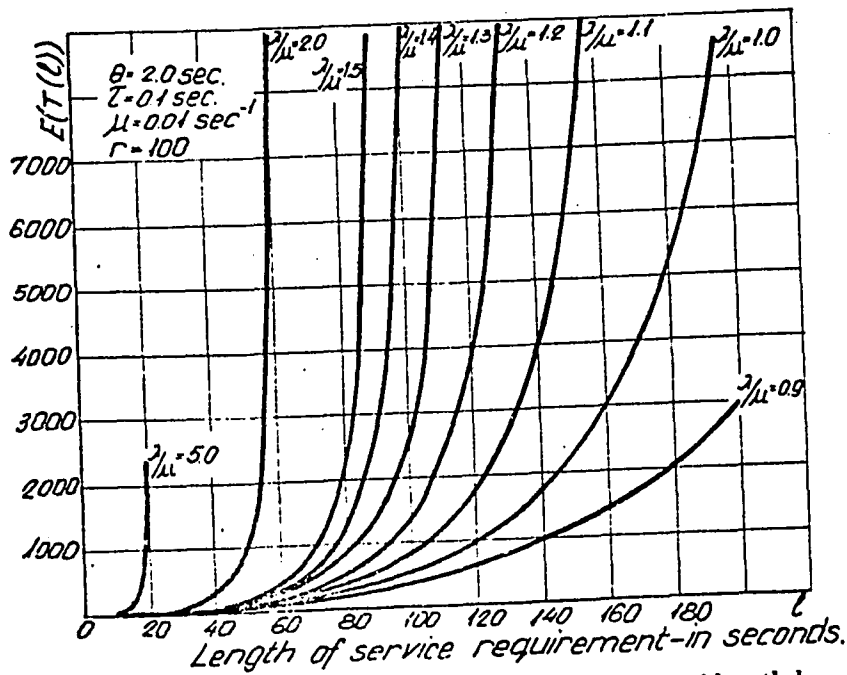
Source: I. Adiri and B. Avi-Itzhak, "A Time-Sharing with many Queues", op. cit. p. 1086

fig. 6:



7.3. 6. Expected total response time for a service requirement of length l as a function of l (for different values of θ).

Expected total response time for a service requirement of length l in seconds.



7.3. 7. Expected total response time for a service requirement of length l as a function of l (for different values of traffic intensity).

Source: I. Adiri and B. Avi-Itzhak, "A Time-Sharing with many Queues", op. cit. p. 1086

disagreement between these results and the analytical results in chapter six. The reader should recall that our simulation results in chapter six disagreed with analytical results for ρ close to 1.0 and especially for units requiring short processing time. In view of these results confirming some of our findings in the previous chapter it will be appropriate to verify the results obtained in this chapter. One should recall that in this chapter we have an over-head factor not previously present.

Hypothesis. The expected response time is properly obtained from the analytical solution.

Simulation Results. The hypothesis was tested by several independent runs of the program TS70 using a procedure described in the previous section. The results are summarized in table 7.4.1. The table shows the mean response time of 12 runs of 200 non-transient observations for each of the points described in the first column. The mean response time for units requiring 0 to 40, 40 to 80, 80 to 120, 120 to 160, and over 160 seconds are given. For each class the mean of the means is given and a 95% confidence interval is shown.

For ease of comparison, the results have been put in graphical form. One can see by comparing figures 7.4.1 and 7.4.2 with figure 7.3.1 that the response time is a function of the quantum size and the service requirement behaves as prescribed by the analytical solution, i.e. that the expected response time obtained from the analytical solution is within the 95% confidence interval. Similarly, by comparing

153
SUMMARY OF THE RESPONSE TIME
ADIRI AND AVI-177HAK MODEL
WITH FINITE NUMBER OF QUEUES

DESCRIPTION		AVERAGE CONDITIONAL RESPONSE TIME DEPENDING ON SERVICE REQUIRED										
		0<S<40, 40<S<80, 80<S<120, 120<S<160, S>160										
L/U=0.85 R=10 Q=10 Q=0.1 1/U=100	MEANS	362 1136	30 31	129 115	238 519	440 1380	1217 4212					
		645 694	26 27	168 117	903 298	1676 543	2265 2487					
		235 360	27 25	127 131	229 275	316 592	826 1007					
		935 348	27 29	124 128	806 238	1513 809	3012 626					
		1465 266	28 28	193 131	634 275	1674 497	4484 594					
		361 203	25 22	86 123	250 252	509 366	1088 1028					
AVER.		855	27	131	402	853	1902					
CONF. INTER		249 1460	26 29	114 147	260 546	521 1183	1027 2776					
L/U=0.85 R=1 Q=1 Q=0.1 1/U=100	MEANS	483 1898	116 418	362 995	354 995	2412 1810	1225 2650					
		1190 1091	293 129	941 373	2482 1090	474 1182	2872 2518					
		372 562	88 89	301 361	429 590	2316 879	989 1283					
		1535 574	225 147	848 485	1547 510	591 974	3587 1227					
				963	188	533	999	1328	2403			
CONF. INTER		500 1425	91 885	339 827	406 1592	699 1958	1229 2858					
L/U=0.85 R=100 Q=10 Q=0.1 1/U=100	MEANS	362 1137	30 34	131 115	226 205	360 619	1267 4761					
		453 740	27 26	170 117	703 230	1360 277	1317 2779					
		234 327	28 25	127 134	216 247	247 396	839 898					
		939 279	27 29	123 141	747 208	208 322	3153 934					
		1359 362	28 29	194 123	341 266	266 511	4476 599					
		347 206	26 32	86 125	240 242	242 631	1106 235					
AVER.		553	28	132	322	608	1871					
CONF. INTER		304 803	26 29	114 149	201 444	362 854	897 2844					
L/U=0.85 R=1 Q=10 Q=0.1 1/U=100	MEANS	357 1053	85 275	261 467	262 1041	465 1423	900 2945					
		1027 591	272 72	745 258	1079 574	1936 692	2185 1644					
		276 462	63 74	185 289	394 410	299 698	750 1063					
		1013 396	194 127	563 349	1104 349	1606 646	2407 806					
		1532 250	314 52	938 173	1319 284	2069 389	3532 549					
		360 225	74 44	170 183	320 259	518 313	889 563					
AVER.		628	137	391	615	893	1518					
CONF. INTER		360 896	74 200	223 540	364 267	473 1312	877 2160					
L/U=0.85 R=10 Q=20 Q=0.1 1/U=100	MEANS	365 1075	34 40	148 110	222 207	358 578	1256 4704					
		464 780	32 31	168 124	700 225	1464 280	1339 2833					
		232 314	31 30	127 122	218 249	299 400	822 911					
		892 313	34 35	133 119	631 211	1360 543	3034 987					
		1367 252	33 31	145 259	336 259	776 445	4449 600					
		346 201	31 29	92 122	248 233	380 345	1081 605					
AVER.		466	33	138	314	588	1882					
CONF. INTER		284 648	31 35	111 166	204 419	329 852	941 2823					
L/U=0.85 R=10 Q=1.0 Q=0.1 1/U=100	MEANS	485 1897	89 282	352 929	360 1791	622 2723	1286 5452					
		1193 1210	190 182	887 350	398 1523	2525 2634	2751 3456					
		1526 906	212 127	865 727	999 1125	2502 2725	3600 3788					
				1202	180	768	1199	2256	3309			
		CONF. INTER		695 1710	110 250	514 991	680 1718	1399 3114	1970 4808			

NOTE: S = LENGTH OF SERVICE; Q = QUANTUM SIZE IN SECONDS; U = SERVICE RATE;
1/U = MEAN SERVICE; R = NUMBER OF QUEUES; L = ARRIVAL RATE;
N = NUMBER OF SOURCES OR THE NUMBER OF USERS;
0 = OVERHEAD; CONF. INTER. = 95% CONFIDENCE INTERVALS.

TABLE 7.4.1

SOURCE: SUMMARY OF MANY RUNS OF THE PROGRAM 1570 WRITTEN BY THE AUTHOR
AND LISTED IN APPENDIX NINE.

figure 7.2.3 with figure 7.3.3, one can see that the analytical results are within the confidence intervals of the simulation results. The second hypothesis cannot be rejected.

. As we have shown by simulation in chapter six and have confirmed in this chapter both by analytical methods and by simulation, in a feedback model, units which need only a few quanta of service will be serviced even though other units with longer requirement may wait very long as the utilization factor approaches 1. The model become similar to an infinite number of feedback models. For example, if r is 100, one could see in figure 7.4.3 and 7.3.3 that r increases the advance for small job is increased. This also confirms our finding in chapter six concerning the expected response time for unit requiring only one quantum.

The difference obtained in the two chapters can be explained by the different approach used in the analytical solution. In the first model there was no overhead factor as in the second. The servicing of the last queue is different. In the last model, since there is a finite number of queue, the last queue is serviced in a Round-Robin manner. There is also a difference in the analytical solutions. In the first model Kleinrock and Coffman assumed that one can break down the waiting time in two parts as explained in chapter six. Furthermore, Kleinrock and Coffman have assumed that even though the arrival process to low priority queues is not Poisson it does not influence the results. The above differences in the solution may explain the difference in the results.

7.

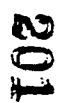
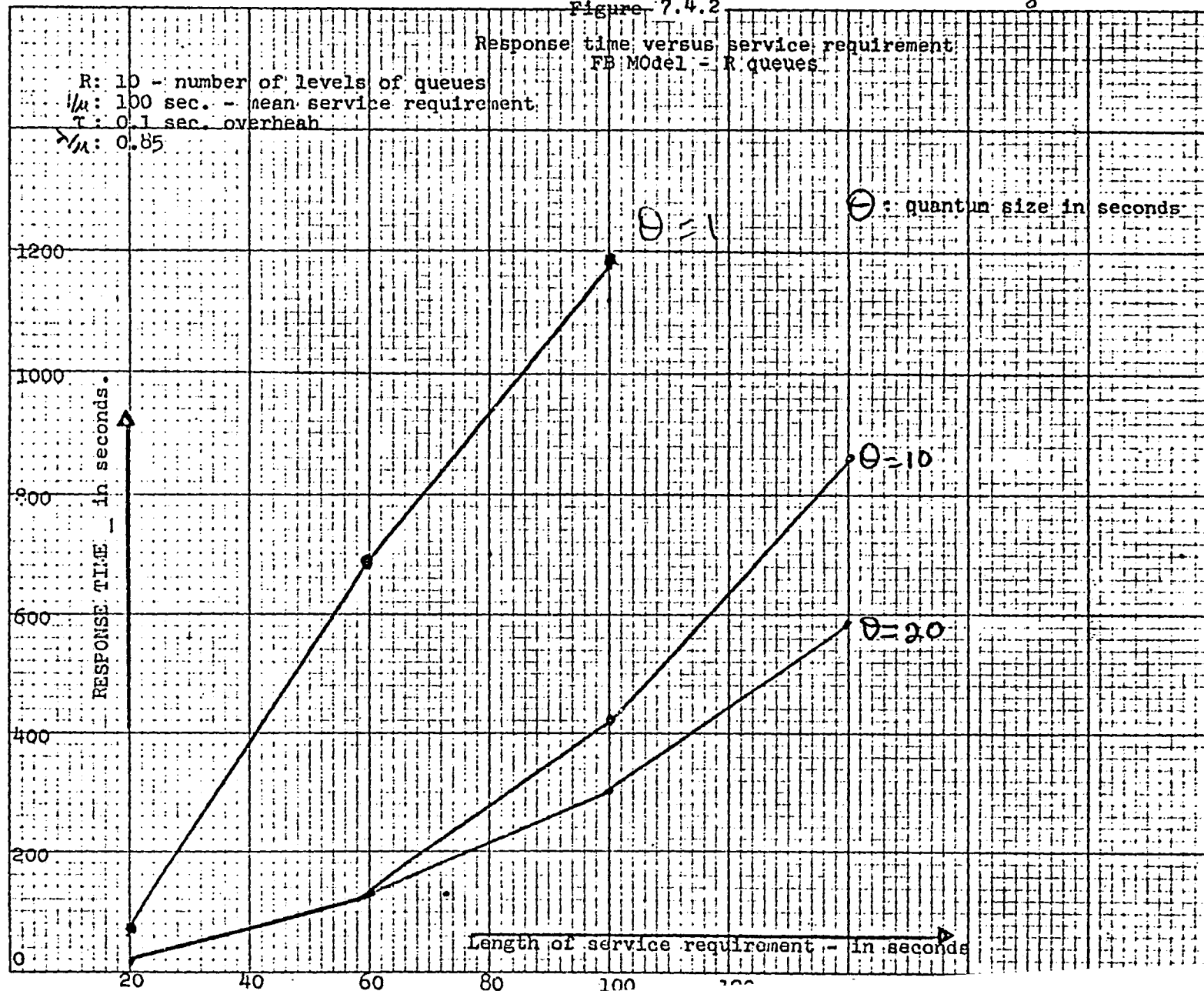


Figure 7.4.2

8



CONCLUSION

In this chapter, the author has reviewed and tested two finite time-sharing models. A model with limited customers and a model with limited feedbacks were analyzed. Having already tested in Chapter Six the sensitivity of similar models, and because of the amount of computer time necessary for each run, we only tested the validity of the analytical solution. We rejected the analytical solution of the model with a limited number of customers for non-steady state conditions, i.e. for utilization greater than 1 where $N\lambda/\mu$ is 1 (which does not include the overhead factor). However, we did not reject the analytical solution in steady-state cases. The effect of increasing the number of users on the system has been shown.

The analytical solution for the model with a limited number of feedbacks was confirmed. We showed, as we did in Chapter Six, that in accordance with the analytical solution in a feedback model the units requiring only a few quanta will get very small response times even if the unit requiring long service time may wait for ever when the system becomes overloaded. We have therefore confirmed that small jobs are not sensitive to preceding jobs as long as the number of queues is large. Thus small jobs will never reach the r^{th} queue where the discipline is Round-Robin.

CHAPTER EIGHT

A MODEL WITH EXTERNAL PRIORITY

The purpose of this chapter is to test by simulation the results of a proposed analytical solution for a model with external priorities. The method used is to describe the model, discuss the published analytical solution, and make simulation experiments on the model to test the analytical solution and to test the effect of a departure from the classical "Poisson assumption".

The concept of priorities in a time-sharing environment was first introduced in a previous chapter. However, in contrast with the previous time-sharing models where the priorities are assigned by the computer either statistically or dynamically, some models have been proposed where priorities are determined before the units get to the computer. The cost of delay may be different for each user or for each class of customers in a time-sharing or real-time system. Therefore, users may be assigned to different priority classes. The priority queuing model is important because of the cost of delaying certain customers. For example, during peak hours, only certain types of jobs or customers may be serviced.

In the previous chapters we have described the contribution to the general queuing models with priorities made by Cobham¹ and Phipps². The first one studied a model with discrete

¹Cobham. "Priority Assignment in Waiting Line Problems". Journal of Operations Research, Soc. of America, Vol. 2, 1959.

²Phipps. "Machine Repair as A Priority Waiting Line Problem", Operations Research, Vol. 4, pp. 76-85, 1956.

external priorities in terms of the expected waiting time for each class. The second, expanding this work, obtained similar results for continuous priorities which could be determined by the length of the required service. Takacs³ studied a model of external priorities when customers are grouped into classes and each class is assigned a priority. He found the expected waiting time in the system for all customers, in a preemptive and non-preemptive model. Kleinrock and Coffman⁴ have studied the distribution of attained service time for customers in different priority groups i.e. the length of time an incompletely serviced customer has spent so far in the service facility. They measured the relative state of partial service received by the customers still in the system. They provided results for both the finite quantum system and for the the system where the quantum size approaches 0. Kleinrock⁵ has shown that in a priority queuing system with Poisson input, with no defection, and with no work loss, the weighted sum of waiting time is a constant, independent of queue discipline. Adiri⁶ analyzed a model with a preemptive priority discipline. He viewed the arrival of high priority customers as a breakdown of the service facility because the low priority unit is removed. He obtained that way the expected total time in the system and expected response time conditional on the length of service required. Recently Chang⁷ compared

³ L. Takacs. "Priority Queues", Operations Research, Vol. 16, No. 1, 1964.

⁴ C. Kleinrock and E. Coffman, "Distribution of Obtained Service in Time-Shared Systems", Journal of Computing Science, (1967) pp. 287-298.

⁵ L. Kleinrock. "A Conversation Law for a Wide Class of Queuing Disciplines", Nav. Res. Lg. Quant. 12 (1965), pp. 181-192.

⁶ I. Adiri. "A Time-Sharing Queue with Preemptive Priority Discipline", Operations Research, Statistics and Economics. No. 7, Haifa, Israel, Dec. 1967.

⁷ W. Chang. "Single Server Queuing Processes in Computing Systems", IBM Systems Journal, Vol. 9, No. 1, (1970)

the efficiency of the two types of priorities and found the first two moments of the waiting time. His solution will be studied in detail in this chapter.

THE MODEL TO BE STUDIED

In this real-time or time-sharing model, the requests are grouped in priority classes depending on the origin. In the simple case, the number of sources is finite and all requests have the same priority. When the computer becomes free, it examines the queue and selects the first request in the highest priority class. One may consider this as a system in which there is a queue for each priority class. There are a large number of queues as in the feedback model analyzed in a previous chapter.

There are two cases of priorities. In a preemptive priority discipline, a unit in service is removed if a user comes in with a higher priority. In the non-preemptive discipline the service of any user cannot be interrupted before either the end of the needed service time or before the end of the quantum. In this model, we will study a solution for an unlimited quantum, i.e. the only time the service can be interrupted is in a preemptive discipline when a unit of higher priority comes into the system. The solutions of both preemptive and non-preemptive priority disciplines are similar. Therefore, we will study the two models together.

Parameters and Assumptions. The following assumptions are made.

1. The input process is a Poisson process with the inter-arrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

where the mean inter-arrival time is $1/\lambda$ seconds and the mean arrival rate is λ , units per second.

2. For the service process an exponential service distribution is assumed.

$$B(r) = \begin{cases} 1 - e^{-\mu r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad 6.1.02$$

where the mean service requirement is $1/\mu$ seconds = a .

3. There is an infinite supply of customers in a priority class.
4. The measure of efficiency is the expected waiting time conditional on the priority number.
5. There is no time lost for setting up, i.e., an interrupted unit carries with it the knowledge of the processing already done to it.
6. There is an infinite number of queues, i.e. there are enough queues so that they all have a priority less than the preceding one or n is large enough so that we need only $n - 1$ queues. This is equivalent to searching in an infinite queue for the highest priority.
7. There is a limited number of origins and at all origin requests have the same priority.
8. There is an infinite quantum of service, i.e. only one pass through the system.

ANALYTICAL SOLUTION

Many authors have presented solutions for a priority queuing model with one server. Chang's⁸ solution will be studied here.

The modeling of a priority system suggested by Chang is a two-step procedure. First, the system is treated as though there were no priority. Second, the effect of priority is taken into account by modifying the results obtained in the first step. The procedure involves Markov Chains. In this case, the concept of imbedded Markov Chains is used because queuing processes are not always Markovian in nature. Kendall proposed to study a system by extracting a set of points at which the Markov property exists⁹. He called those points, regeneration points. For example, the instants at which customers depart from the system are regenerations points. This procedure involves extracting a discrete time Markov chain imbedded in the continuous time process.

In the preemptive case, the presence of non priority users does not delay the high priority users. The presence of users of priority greater than k does not influence the distribution of waiting time of users with priority less than k . We may then consider the users of priority less than k as non-present in the system when we study users of priority greater than k . In order to obtain the distribution

⁸ Chang, op.cit. p. 37-71

⁹ The concept of imbedded Markov Chains is discussed in appendix fifteen, It is taken from:
D.G. Kendall, "Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of Imbedded Markov Chain The Annals of Math. Stat. No.24, (1963) pp. 338-354.

of the waiting time of units in class k , we consider a modified queuing process with the following characteristics:

1. The input is Poisson with parameter

$$\lambda_k = \sum_{i=1}^K \lambda_i$$

2. The service time distribution is weighted by the input rates.

The generating function is obtained using the imbedded Markov chains. Then the Laplace-Stieltjes transform of the waiting time is found for the modified queuing process. The expected waiting time $W_k^{(1)}$ for the modified queuing process is obtained by differentiating the Laplace-Stieltjes transform. The waiting time distribution $W_k^*(x)$ for users of priorities k is obtained by considering two times. First, there is waiting behind users of priority less than k . Second, there is waiting behind users of priority equal to k . The Laplace-Stieltjes transform is found in a similar way. The differentiation gives the expected waiting time $W_k^{(1)}$ units of priority k .

In a non-preemptive service discipline, the service time of a user is not interruptible. Therefore, if a high priority unit comes in it may have to wait until a low priority unit completes its processing. On the other hand, low priority users may be serviced right away if the computer i.e. the single server, is free. An approach similar to the one given above is used to obtain $W_k^{(1)}$, i.e. the expected waiting time of a unit of priority k in the modified queuing process, and $W_k^{(1)}$ i.e. the expected waiting time of a unit of priority k .

The following results have been obtained by Chang for the waiting time using the steps given in appendix fifteen¹⁰.

An expression is given for the two first moments

where $W_k^{*(n)}$ is the n^{th} moment of the waiting time for the unit of priority k in the modified queuing process,

$W_k^{(n)}$ is the n^{th} moment of the waiting time for the unit of priority k ,

$W_k^{(1)}$ is the expected waiting time.

CASE 1. Preemptive-resume discipline

$$W_k^{*(1)} = \frac{\Lambda_k a_k^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]}$$

$$W_k^{*(2)} = \frac{\Lambda_k a_k^{(3)}}{3[1 - \Lambda_k a_k^{(1)}]} + \frac{\Lambda_k^2 [a_k^{(2)}]^2}{2[1 - \Lambda_k a_k^{(1)}]^2}$$

where $a_k^{(r)}$ is the r^{th} moment of the weighted service time distribution

CASE 2 Non-Preemptive discipline

$$W_k^{*(1)} = \frac{\Lambda_k a_k^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]}$$

$$W_k^{*(2)} = \frac{\Lambda_k a_k^{(3)}}{3[1 - \Lambda_k a_k^{(1)}]} + \frac{\Lambda_k a_k^{(2)} \Lambda_k a_k^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]^2}$$

Finally, we compute the waiting-time moments from (s) by differentiating as follows:

$$W_k^{(1)} = W_k^{*(1)} (1 + \Lambda_k - 1 d_{k-1}^{(1)}) = \frac{W_k^{*(1)}}{1 - \Lambda_k - 1 d_{k-1}^{(1)}}$$

and

$$\begin{aligned} W_k^{(2)} &= W_k^{*(2)} [1 + \Lambda_k - 1 d_{k-1}^{(1)}]^2 + W_k^{*(1)} \Lambda_k - 1 d_{k-1}^{(2)} \\ &= \frac{W_k^{*(2)}}{[1 - \Lambda_k - 1 d_{k-1}^{(1)}]^2} + \frac{W_k^{*(1)} \Lambda_k - 1 d_{k-1}^{(2)}}{[1 - \Lambda_k - 1 d_{k-1}^{(1)}]^3} \end{aligned}$$

where $d_k^{(n)}$ is the busy period moment such that:

$$d_{k-1}^{(1)} = \frac{a_{k-1}^{(1)}}{1 - \Lambda_{k-1} a_{k-1}^{(1)}}$$

$$d_{k-1}^{(2)} = \frac{a_{k-1}^{(2)}}{[1 - \Lambda_{k-1} a_{k-1}^{(1)}]^3}$$

Chang also gives the following examples:

Where there are 3 sources

$\lambda_1 = \lambda_2 = \lambda_3 = 0.3$ calls/seconds
 $a_1 = a_2 = a_3 = 1$ second exponentially distributed. The results obtained are as follows:

The weighted service-time moments are the following:

$$a_1^{(1)} = a_2^{(1)} = a_3^{(1)} = 1 \text{ second,}$$

$$a_1^{(2)} = a_2^{(2)} = a_3^{(2)} = 2 \text{ seconds.}$$

Let the busy periods for the priority system be defined as follows:

$d_1^{(1)}$ is the busy period of the server for the first priority class,

$d_2^{(1)}$ is the busy period of the server for the first and second priorities,

$d_3^{(1)}$ is the busy period of the server for all three classes.

In this example, the mean busy periods for the three priority classes are as follows:

$$d_1^{(1)} = \frac{1}{1 - 0.3} = 1.429 \text{ seconds}$$

$$d_2^{(1)} = \frac{1}{1 - 0.3 - 0.3} = 2.5 \text{ seconds}$$

$$d_3^{(1)} = \frac{1}{1 - 0.3 - 0.3 - 0.3} = 10 \text{ seconds}$$

The mean waiting times for the two cases can now be calculated.

Case 1 Preemptive-resume discipline

Mean waiting time for the first priority:

$$W_1^{*(u)} = \frac{0.3 \times 2}{1 - 0.3} = 0.855 \text{ seconds}$$

$$W_1^{(u)} = W_1^{*(u)} = 0.855 \text{ seconds}$$

Meaning waiting time for the second priority:

$$W_2^* = \frac{0.6 \times 2}{1 - 0.3 - 0.3} = 3 \text{ seconds}$$

$$W_2^{(u)} = W_2^{*(u)}(1 - \lambda_1 \alpha_1^{(u)}) = 3 \times 1.436 = 4.31 \text{ seconds}$$

Mean waiting time for the third priority:

$$W_3^{*(u)} = \frac{0.9 \times 2}{1 - 0.3 - 0.3 - 0.3} = 18 \text{ seconds}$$

$$W_3^{(u)} = \frac{W_3^{*(u)}}{1 - \lambda_1 \alpha_1 - \lambda_2 \alpha_2} = \frac{18}{0.4} = 45 \text{ seconds}$$

Case 2 Nonpreemptive discipline

Mean waiting time for the first priority:

$$W_1^* = \frac{0.9 \times 2}{1 - 0.3} = 2.57 \text{ seconds}$$

$$W_1^{(u)} = 2.57 \text{ seconds}$$

Mean waiting time for the second priority:

$$W_2^* = \frac{0.9 \times 2}{1 - 0.3 - 0.3} = 4.5 \text{ seconds}$$

$$W_2^{(u)} = \frac{4.5}{1 - 0.3} = 6.4 \text{ seconds}$$

Mean waiting time for the third priority:

$$W_3^{*(1)} = \frac{0.9 \times 2}{1 - 0.3 - 0.3 - 0.3} = 18 \text{ seconds}$$

$$W_3^{(1)} = \frac{18}{1 - 0.3 - 0.3} = 45 \text{ seconds}$$

The mean waiting times for the first and second priorities in Case 2 are higher than those for Case 1 because the server in Case 2 completes the execution of a lower-priority call in service before ^{it is} interrupted to serve a higher-priority call. If one considers the same model with no priority, the waiting time would be 9.0 seconds.

In the above discussion, the priority model has been limited to a real-time or to a time-sharing environment with infinite quantum size. The reader should make a distinction between this model where the priorities are assigned by the origin of the request and the model studied earlier in this dissertation where the priority is dependent on the length of service.

One may wonder why there is the same waiting time for the units of the third priority in both the preemptive case and the non-preemptive. One may also question the fact that the second priority units have greatly differing times.

One should remember that a two-step procedure has been used to solve the model and that the outcome of the first step is modified to take into account the type of priority. It is therefore worthwhile to test the following hypotheses.

HYPOTHESES

- a. The solution proposed above is valid.
- b. The users always do better in the preemptive case than in the non-preemptive case.
- c. The arrival distribution has no effect on the the response time.

SIMULATION RESULTS

Testing the first Hypothesis. In order to test the model the author wrote the program PR10 listed in appendix eight. The model was run with 1000 transient arriving units followed by 1000 steady state units. It was found that the results were the same as a 50 transient period and 100 steady state units.

The results of 36 runs are summarized in table 8.2.1. Confidence limits are given in table 8.2.2. An example of the actual output is shown in table 8.2.3. For each queue, the program gives the usual statistics, i.e. the number of entries, the maximum content, the average waiting time, and the average waiting time for the units which wait. The distribution of the time spent in the system is divided into two sections. Section "Xtml" gives the distribution of the response time for the units requiring more than 5 seconds while "Xtime" gives the distribution of waiting time for all units.

WAITING TIME IN THE QUEUE CHANG'S MODEL WITH EXTERNAL PRIORITY								
	NONPREEMPTIVE				PREEMPTIVE			
W1	1075	1034	1072	1214	250	397	190	566
	1072	925	1458	1169	624	350	281	600
	1196	842	1412	1102	534	50	278	298
W2	2673	2584	3221	2327	1433	1091	1736	1664
	3046	1769	1792	1967	1072	1816	651	1912
	2218	962	4834	2332	1004	3046	1370	1387
W3	20935	5403	11005	3067	3722	5578	5374	11764
	12368	2212	4363	8442	8525	2266	12278	4484
	4341	1863	12624	12954	1924	7339	4880	11754
WA	9014	2977	4682	2297	1780	2484	2421	4199
	4634	1686	2727	4526	5221	1508	4039	2178
	2480	1163	5865	5510	1152	3641	2332	4367
FIFO	2967	5595	4317	3402				
	5439	1729	7636	2670				
	20759	9580	5110	3410				

NOTE:

ARRIVAL RATE = 3 CALLS PER SECOND AT EACH OF THE THREE SOURCES.

WA = AVERAGE OF WAITING I.E. $(W1 * N1 + W2 * N2 + W3 * N3) / (N1 + N2 + N3)$,

W1 = WAITING TIME FOR UNITS WITH HIGHEST PRIORITY.

SERVICE TIME = 1 SECOND.

TABLE: 8.2.1

SOURCE: SUMMARY OF THE OUTPUT OF THE PROGRAM PRIO WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX EIGHT.

CONFIDENCE LIMITS FOR WAITING TIME				
	MEAN OF 12 RUNS	INTERVALS		ANALYTICAL RESULTS
		95%	99%	
NONPREEMPTIVE				
PRIORITY 1	1128	1019 - 1237	974 - 1232	2570
PRIORITY 2	2472	1861 - 3082	1612 - 3331	4500
PRIORITY 3	8298	4620 - 11975	3122 - 13473	45000
AVERAGE A	3963	2566 - 5360	1997 - 5929	11690
PREEMPTIVE				
PRIORITY 1	352	223 - 481	170 - 534	855
PRIORITY 2	1516	1128 - 1904	970 - 2062	4310
PRIORITY 3	6657	4324 - 8990	3374 - 9940	45000
AVERAGE A	2937	2112 - 3763	1775 - 4099	7672
FCFS				
NO PRIORITY	6409	2424 - 10933	710 - 12107	8100

NOTE: THE TIME IS EXPRESSED IN MILLISECONDS.

TABLE: 8.2.2

SOURCE: OUTPUT OF THE PROGRAM FINCON01 WRITTEN BY THE AUTHOR AND LISTED IN APPENDIX SEVEN.

UTILIZATION OF THE CPU

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN	SEIZING TRANS. NO.	PREEMPTING TRANS. NO.
CPU	.855	1000	958.847		

WAITING TIME FOR THE CPU

QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	SAVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
1	5	.327	333	49	14.7	1102.669	1292.918		
3	6	.687	330	60	18.1	2332.772	2851.168		
4	17	3.908	338	35	10.3	12954.820	14451.253		

SAVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

TIME SPENT IN THE SYSTEM

TABLE ENTRIES IN TABLE	XTM1 575	MEAN ARGUMENT 7235.507	STANDARD DEVIATION 9936.000	SUM OF ARGUMENTS 4160418.000	NON-WEIGHTED	
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-.728
500	0	.00	.0	100.0	.069	-.677
1000	53	9.21	9.2	90.7	.138	-.627
1500	55	9.56	18.7	81.2	.207	-.577
2000	70	12.17	30.9	69.0	.276	-.526
2500	53	9.21	40.1	59.8	.345	-.476
3000	45	7.82	47.9	52.0	.414	-.426
3500	28	4.86	52.8	47.1	.483	-.374
4000	37	6.43	59.3	40.6	.552	-.325
4500	23	3.99	63.3	36.6	.621	-.275
5000	28	4.86	68.1	31.8	.691	-.224
5500	16	2.78	70.9	29.0	.760	-.174
6000	11	1.91	72.8	27.1	.829	-.124
6500	10	1.73	74.6	25.3	.898	-.074
7000	9	1.39	75.9	24.0	.967	-.023
7500	9	1.56	77.5	22.4	1.036	.026
8000	7	1.21	78.7	21.2	1.105	.076
8500	5	.86	79.6	20.3	1.174	.127
9000	2	.34	79.9	20.0	1.243	.177
9500	3	.52	80.5	19.4	1.312	.227
10000	5	.86	81.3	18.6	1.382	.278
10500	2	.34	81.7	18.2	1.451	.328
11000	1	.17	81.9	18.0	1.520	.378
11500	2	.34	82.2	17.7	1.589	.429
12000	2	.34	82.6	17.3	1.658	.479
12500	1	.17	82.7	17.2	1.727	.529
13000	3	.52	83.3	16.6	1.796	.580
13500	3	.52	83.8	16.1	1.865	.630
14000	2	.34	84.1	15.8	1.934	.680
OVERFLOW	91	15.82	100.0	.0		
AVERAGE VALUE OF OVERFLOW		27609.96				

TABLE: 8.2.3

Sample output of the Program PR10 written by the author and listed in Appendix Eight.

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	45	4.49	4.4	95.5	-.000	-1.135
10	241	24.09	28.5	71.4	.239	-.664
20	118	11.79	40.3	59.6	.478	-.592
30	92	9.19	49.5	50.4	.717	-.320
40	63	6.29	55.8	44.1	.956	-.049
50	72	7.19	63.0	36.9	1.195	.222
60	54	5.39	68.4	31.5	1.435	.494
70	57	5.69	74.1	25.8	1.674	.765
80	39	3.89	78.0	21.9	1.913	1.037
90	44	4.39	82.4	17.5	2.152	1.309
100	172	17.19	99.6	.3	2.391	1.580
110	0	.00	99.6	.3	2.630	1.852
120	0	.00	99.6	.3	2.870	2.124
130	0	.00	99.6	.3	3.109	2.395
140	0	.00	99.6	.3	3.348	2.667
150	0	.00	99.6	.3	3.587	2.938
160	0	.00	99.6	.3	3.826	3.210
170	1	.09	99.7	.2	4.066	3.482
180	0	.00	99.7	.2	4.305	3.753
OVERFLOW	2	.19	100.0	.0		
AVERAGE VALUE OF OVERFLOW		235.00				

TABLE XTIME
ENTRIES IN TABLE
1000

MEAN ARGUMENT
6764.949

STANDARD DEVIATION
10064.000

SUM OF ARGUMENTS
6764952.000

NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-.672
200	36	3.59	3.5	96.3	.029	-.652
400	44	4.39	7.9	92.0	.059	-.632
600	46	3.99	11.9	88.0	.088	-.612
800	49	4.89	16.8	83.1	.118	-.592
1000	43	4.79	21.6	78.3	.147	-.572
1200	40	3.99	25.6	74.3	.177	-.552
1400	37	3.69	29.3	70.5	.206	-.533
1600	35	3.49	32.8	67.1	.236	-.513
1800	42	4.19	37.0	62.9	.266	-.493
2000	37	3.69	40.7	59.2	.295	-.473
2200	27	2.69	43.4	56.5	.325	-.453
2400	33	3.29	46.7	53.2	.354	-.433
2600	33	3.29	50.0	49.9	.384	-.413
2800	22	2.19	52.2	47.7	.413	-.393
3000	22	2.19	54.4	45.5	.443	-.374
3200	17	1.69	56.1	43.8	.473	-.354
3400	10	.99	57.1	42.8	.502	-.334
3600	20	1.99	59.1	40.8	.532	-.314
3800	18	1.79	60.9	39.0	.561	-.294
4000	23	2.29	63.2	36.7	.591	-.274
4200	16	1.59	64.8	35.1	.620	-.254
4400	15	1.49	66.3	33.6	.650	-.234
4600	10	.99	67.3	32.6	.679	-.215
4800	14	1.39	68.7	31.2	.709	-.195
5000	14	1.39	70.1	29.8	.739	-.175
5200	11	1.09	71.2	28.7	.768	-.155
5400	7	.69	71.9	28.0	.798	-.135
5600	4	.39	72.3	27.6	.827	-.115
OVERFLOW	276	27.59	100.0	.0		
AVERAGE VALUE OF OVERFLOW		19148.14				

TABLE PASS
ENTRIES IN TABLE
1000

MEAN ARGUMENT
1.000

STANDARD DEVIATION
.000

SUM OF ARGUMENTS
1000.000

NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	-.000	-.000
1	1000	100.00	100.0	.0	1.000	-.000

REMAINING FREQUENCIES ARE ALL ZERO

MARTIN 2 4

Source: ibid

219

RESPONSE TIME FOR A PRIORITY MODEL						
PRIORITY	TIME				MEAN	95% CONFIDENCE LIMITS
NONPREEMPTIVE	5483	5851	1795	3240	4789	3411 - 6168
	10195	3984	6312	3361		
	5169	2614	3931	5410		
PREEMPTIVE	3565	2780	5205	2161	4007	3290 - 4725
	4375	3991	5843	3366		
	4475	2614	5507	3210		
FCFS	9477	6820	5275	3244	6738	3729 - 9926
	2928	3312	8128	2667		
	4437	10397	4771	19329		

TABLE: 8.2.5

SOURCE: OUTPUT OF THE PROGRAM PR10 AND FINCON01 WRITTEN BY
AUTHOR AND LISTED IN THE APPENDICES.

One can see from table 8.2.2 that there is not enough evidence to reject the analytical solution in the case of the First Come First Served Model (FSFC). However, they do not yield the same results in the case of the priority model. The same program was used in both cases.

Looking at table 8.2.2, the reader can easily see that the results obtained by Chang are outside the confidence intervals. It is easy to reject the first hypothesis. There is a difference between the analytical results and the simulation results. The simulation yields a much smaller time. Chang had suggested that in the case of a non-preemptive priority the average waiting time would be 17 seconds, which is almost twice as much as in the FSFC model. It would be 16.7 in the preemptive case which is much more than the 9 seconds waiting time in the FCFS model.

The difference may be explained by the approximation procedure that was used. The system, although not Markovian in nature, was solved as a Markovian process by extracting points at which the Markov property exists. The model is solved as though there were no priority. The results are adjusted for the priority effect. Furthermore, a modified queuing model is used where the service time distribution is weighted by the input rates.

Testing the Second and the Third Hypotheses. The results shown in table 8.2.5 seem to sustain Chang's results that the average users do better in the preemptive case. In the latter case the number of zero waiting is increased. In the case where there is no overhead, it is to the advantage of all to have units preempting others. If one measures the total response time one can see from figure 8.2.5 that the response time is less for the preemptive case. However, the difference is not statistically significant.

Looking at the above results, one may want to know whether the queue discipline has an important effect on the response time. It is also important to determine whether the arrival distribution has a significant effect on the response time. It may be easier and cheaper to do sensitivity analysis on the assumptions to determine the effect of a departure from the "Poisson" assumption than to field search for the exact distribution.

In order to test the effect of the queue discipline, of the arrival pattern, and of the interaction, 12 different runs were made for each of the 8 cells of the table 8.2.6. The author used four arrival patterns which are by nature very much different from one another.

An analysis of variance was performed as shown in table 8.2.7. The results indicate that no effect is made on the response time by the type of priority, the arrival pattern, or the interaction. If the arrival pattern does not have a significant effect it implies that it is not worthwhile to search for the true arrival pattern if the units are going to be serviced according to external priorities.

RESPONSE TIME FOR THE PRIORITY MODEL																
ARRIVAL DISTRIBUTION																
PRIORITY	POISSON				UNIFORM				CONSTANT				NORMAL			
PREEMP	3565	2780	5205	2161	4604	3144	10940	5132	3857	4283	4826	3347	4084	6092	3579	3581
	4375	3991	5843	3361	3832	5702	2624	8439	3411	3114	3903	3684	4453	2368	5863	2671
	4475	2614	5507	3213	3708	5324	3103	7997	5149	6001	5206	7191	5107	3150	10392	4224
NON- PREEMP	6797	4449	3981	5177	4545	2464	10750	5123	3981	5034	3319	4016	4029	3501	6617	3763
	4488	5620	6228	2840	3594	5478	2686	8710	3379	5403	3224	2856	3442	4845	2475	5921
	7545	4133	10542	4238	3511	5122	3056	8137	5193	3675	7521	6068	2947	4924	3070	10621

TABLE: 8.2.6

SOURCE: OUTPUT OF 72 RUNS OF THE PROGRAM PR10 WRITTEN BY THE AUTHOR
AND LISTED IN APPENDIX EIGHT.

EFFECT OF THE ARRIVAL DISTRIBUTION ON THE RESPONSE TIME PRIORITY MODEL.

ANOVA--LEAST SQUARES SOLUTION

SOURCE	SS	DF	MS	F
A	4073728.000	1.	4073728.000	0.674
S(A)	132963072.000	22.	6043776.000	
B	9130496.000	3.	3043498.000	0.881
AB	10914304.000	3.	3638101.000	1.054
BS(A)	227895808.000	66.	3452966.000	
GRAND MEAN = 4774.145				

ANOVA--UNWEIGHTED SOLUTION

SOURCE	SS	DF	MS	F
A	4072704.000	1.	4072704.000	0.674
S(A)	132963072.000	22.	6043776.000	
B	9128450.000	3.	3042816.000	0.881
AB	10916738.000	3.	3638912.000	1.054
BS(A)	227895808.000	66.	3452966.000	
GRAND MEAN = 4774.141				

CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B

	1	2	3	4
1	3924.500	5295.164	4422.664	4630.332
2	5503.164	5264.664	4471.582	4681.082

A = QUEUE DISCIPLINE, B = ARRIVAL DISTRIBUTION; BOTH ARE FIXED FACTORS.

TABLE: 8.2.7

SOURCE: OUTPUT OF THE PROGRAM FINAOV LISTED IN APPENDIX FIVE.

CONCLUSION

This chapter has described and tested a priority model in order to compare two basic priority cases: preemptive priority and non-preemptive priority.

The waiting time obtained from the simulation runs is significantly different from the one suggested by the analytical solution. Furthermore, the results have shown that the type of priority and the arrival distributions do not have a significant effect on the response time. However, we have sustained the conclusion that users do better in a preemptive priority model.

CONCLUSION

In this dissertation, computer simulation has been used successfully to show that some real-time and time-sharing models do not behave as prescribed by the mathematical expressions. It has also been shown that it is worthwhile to do sensitivity analysis on the arrival assumptions, since it has been found that in some models the arrival pattern does not have an effect on the response time, whereas in others it does. In all cases, the systems have been measured from the point of view of the user, so that the measure of efficiency, instead of being the congestion, was with the expected waiting or the expected response time.

Many authors in recent years have expressed the need to have some of the approximation procedures verified. They also have expressed the urgent need to obtain solutions for models with non-Poisson processes. Until now, most authors would either intuitively describe the reasons why the arrival should be Poisson, or say that a solution is possible only for Poisson processes. Sometimes they would intuitively show that the solution should not be much different for non-Poisson processes.

The main contribution of this dissertation beyond previous studies has been to verify some analytical procedures, to develop solutions for non-Poisson processes, and to study formally the effect of a departure from the Poisson assumption. It has been shown that some of the analytical procedures were not valid, models have been identified where

the solution was not rejected and analytical procedures have been developed which were not shown to be invalid under simulation experiments. By analysis of variance we also have pointed out the models in which the response time is the same for different arrival distributions.

In Chapter Four, it was shown that the analytical solution for a multi-stage real-time model was not valid. By analysis of variance we arrived at the original conclusion that the arrival distribution has no significant effect on the response time. In Chapter Five, we developed an analytical solution for a direct access model, and simulation experiments did not reject the analytical solution. Furthermore, it was shown that this model was sensitive to the arrival distribution and to the interaction of the arrival distribution with the number of disks. It was also explained how the simulation can easily handle any arrival distribution. In Chapters Six and Seven, different time-sharing models, such as Feedback, Round-Robin, and Shortest Job First, were tested. Some analytical solutions given by Adiri and Avi-Itzhak were not rejected. It was also shown that in all time-sharing models the arrival pattern has a significant effect on the response time but that there is no interaction between the arrival distribution and the utilization of the service station. In Chapter Eight, the analytical results for a model with external priorities were tested. It was shown that the results of the simulation were significantly different. It was also found, as in Chapter Four, that the arrival pattern does not have a significant effect on the response time.

The dissertation has also shown that simulation is useful in solving queuing problems. Simulation models being very flexible, it is easy to do sensitivity analysis on some parameters of the model to determine their effect. This is the most important aspect of simulation from the managerial point of view. Every model deals with only some aspects of the real world; it is therefore a simplification of the real world where many of the parameters are neglected or assumed. Simulation can be used directly to determine the effect of neglecting certain parameters or to see what is the effect of a departure from the assumed values.

Simulation has also been shown to be useful in the study of a system composed of sub-systems. It may be easy to use analytical tools to solve all the parts of the system. However, there is no assurance that the whole is equal to the sum of the parts. Simulation can be used to analyze the interaction of subsystems or to solve the system as a whole.

Problems For Further Study. Further research is needed to determine the classes of models where a departure from the Poisson assumption does not have a significant effect on the response time. Studies should be made with more types of arrival distributions. The same analysis should also be done for several service distributions in order to determine whether or not the type of service has a significant effect on the response time. Further studies should also repeat the same analysis for interaction effect between the arrival distribution and the service distribution. As mentioned above this dissertation was concerned only with the customer's point of view, i.e. the waiting time and the response time. Further research should be done to ascertain whether or not the arrival pattern has an effect on the congestion.

Simulation is expensive and the cost is directly related to the size of each of the many runs. A technique is required to remove the effect of autocorrelation and to stop the run when the level of confidence is attained. In this dissertation, the problem was circumvented by making many runs with different random numbers series and by taking the mean of the means, but this procedure may increase the cost of the experiments. This procedure leads to the question of trade-off between the number of repetitions and the number of observations per run. Although, for each model, we have observed a large number of transactions in the system, usually 200 per run, we had a small sample size, i.e. 12 repetitions. Future research should determine whether the need to have a long run to ensure independence is as important as the need to have a large number of repetitions in order to apply the Central Limit Theorem. Further studies should also be made to determine whether it would be more efficient to estimate the autocorrelation function rather than the procedure followed in this dissertation. Future research on computer simulation should certainly be oriented toward developing efficient stopping rules and statistical techniques for validation of the results rather than toward the development of more computer languages.

The procedure used to remove the autocorrelation effect resulted in small samples. There was therefore, not enough evidence to reject some of the analytical solutions because of the high variation in the response time. It would be useful to repeat more experiments in order to get more reliable results. It would also be meaningful to complement

this research by repeating the experiments for other parameter values. Results that were not rejected for specific parameters may be rejected for other values.

An important cost generating factor in this research was the elimination for each run of the initial bias. More research should determine whether a procedure of dividing a long run into segments would be preferable, even if there is correlation between segments. A procedure may even be developed to determine by program the exact number of observations to ignore in each run.

In this dissertation, it was pointed out that one of the errors introduced in some of the analytical solutions was the approximation made in estimating the output distribution of a complex queuing system. It would be necessary to go one step further in this research and to measure by simulation the shape of the output distribution. It would also be useful to measure any other area where approximations have been used.

Simulation Versus Analytical Tools. In this dissertation we have shown that some analytical solutions were not valid for certain models. However, it is not obvious whether simulation experiments or analytical tools would be more appropriate in all cases. We have shown that in complex queuing systems made up of subsystems it was more appropriate to use simulation in order to properly estimate the mean and the variance of the response time of the whole system. We have explained that in simulation it is not necessary to measure the output of each stage in a multi-stage or feedback model. Instead, we follow a unit throughout the model from the time it joins the first queue to the end of the last service station.

No general conclusion on the question of when to use or not to use analytical techniques has been reached. The management scientist is always faced with limited resources in solving a practical problem. He has to allocate resources in the most efficient manner to achieve the objective of his effort. The decision-maker must always consider that the payoff from a model must offset the development and solution expenses.

As we have pointed out in this dissertation, computer simulation is an inductive procedure as opposed to analytical tools, which are deductive. Being experimental, it requires a great amount of computer time to generate enough observations to have reliable estimates of parameter values. The cost of an experiment increases with the degree of precision desired.

Although, as we have shown, some analytical solutions are not correct because of approximation procedures, it does not imply that they can never be used. Analytical solutions of simple models can be more accurate because there is less need to make approximations in between stages or in between subsystems. Analytical techniques can certainly be used in complex models if one is ready to accept the errors they generate i.e. if the value of more precision does not justify the cost of using simulation or if one does not have the resources to use computer simulation. It is possible that analytical procedures may be improved in cases where they are imprecise or they may be developed in cases where they do not exist. However, one may improve greatly analytical solutions of queuing models by using simulation to find the exact distribution of the output at each stage ; this is the main source of errors introduced in solutions of complex cases, as we have shown in this dissertation. This implies that one may find an optimum mix of the two techniques in allocating available

resources. Analytical tools and simulation should therefore be seen, not as competing but rather as complementary, tools

Simulation and Optimality. The decision maker must realise that the output of a simulation run does not provide him with an optimum solution to his problem. It simply represents the response of the system under a single chosen set of values for the parameters, controlled variables, or assumptions. If a manager wants to find the optimum configuration of a computer system, or any other system, he must run the model with different conditions. Only by going from one configuration to the next after each series of sampling experiments can he find a configuration or a series of configurations which are optimum.

By optimum we do not mean the best solution in an absolute sense, but we mean the best solution that can be attained with limited resources and limited time. Searching for the optimum by simulation is like searching to find the peak of a mountain. We start from a known point and we change parameters until we reach a point where our measure of efficiency starts to decrease. However, there may be more than one peak.

Sensitivity Analysis and Simulation. Both of these methods try to answer the same question: "What if". However, they are not synonymous. The first can be applied to deterministic models such as linear programming, the second implies, as we have defined, a stochastic model. In this dissertation we have formally joined the two in testing the sensibility of the system to the arrival assumptions. In the same way, when solving a deterministic linear model, a manager may test the sensitivity of the model to an error in estimating a constraint.

The decision maker may use in the real world sensitivity analysis as we have done in this dissertation to see how possible changes, errors, or even ignorance about a factor may affect the system. If the system is not sensitive to the factor being tested there is no need to purchase more information about that factor. For example, in this dissertation some models were not sensitive to the arrival pattern. This implies that there was no need to field test to find the exact shape of the distribution. If the system is sensitive to a factor, a statistical decision model may be developed.

The need for sensitivity analysis varies with the degree of uncertainty. It is more important to test the sensitivity of factors that are less certain. However, one must realize that the decision maker is never completely free from uncertainty.

The decision-maker of the future will profit from simulation by evaluating the realness of the assumptions upon which the usefulness of the model rests. However, the most important requirement for the successful decision maker living in the real world and facing real problems is to be able to face changes and to analyze feedback information. Therefore, the model must continually evolve to adapt itself to the managers' understanding of the real world. Computer simulation is not designed to replace management judgment but to provide a better understanding of the interaction of several factors affecting a decision. It should provide better information upon which to base a decision.

*** APPENDIX ONE ***

PROGRAM RAIOVI - RAIOVI ACCESS ONE CHANNEL
AUTHOR : PROSPER M. BERNARD

// JOB RAIOVI DIRECT ACCESS

ALLOC FI=OK,F2=OK

// ASSIGN SYS000,X'131'

* MOUNT A WORK PACK ON 131

// DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD

// EXTENT SYS000,11111,1,0,0310,610

// DLBL INTER0,'SYSTEM WORK FILE NO. 0',68/032,SD

// EXTENT SYS000,11111,1,0,0310,610

// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD

// EXTENT SYS001,11112,8,1,2920,240,7

// DLBL SIMIN01,'SYSTEM WORK FILE NO. 1',67/365,SD

// EXTENT SYS001,11112,8,1,2920,240,7

// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD

// EXTENT SYS002,11112,8,1,2929,180,12

// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD

// EXTENT SYS002,11112,8,1,2929,180,12

// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD

// EXTENT SYS003,11112,8,1,2933,180,12

// DLBL REPR001,'SYSTEM WORK FILE NO. 3',67/365,SD

// EXTENT SYS003,11112,8,1,2933,180,12

// END

// EXEC DAP01

SIMULATE

***** ACCESSING IBM 2311 DISK STORAGE DEVICES *****

***** ONE CHANNEL ONLY IN THE SYSTEM *****

```

0.25,1/0.50,2/0.75,3/1.0,4
15 FUNCTION R41,D5 5 DISKS RA100105
0.2,1/0.4,2/0.6,3/0.8,4/1.0,5
16 FUNCTION R41,D5 6 DISKS RA100110
0.166,1/0.233,2/0.49,3/0.664,4/0.830,5/1.0,6
17 FUNCTION R41,D7 7 DISKS RA100110
0.143,1/0.286,2/0.429,3/0.572,4/0.715,5/0.858,6/1.0,7
18 FUNCTION R41,D8 8 DISKS RA100109
0.125,1/0.250,2/0.375,3/0.500,4/0.625,5/0.750,6/0.875,7/1.0,8
19 FUNCTION R41,D9 9 DISKS RA100110
0.111,1/0.222,2/0.333,3/0.444,4/0.555,5/0.666,6/0.777,7/0.888,8/1.0,9
8 FUNCTION R41,D2 1 CHANNELS RA100110
0.5,10/1.0,10
9 FUNCTION R41,D2 2 CHANNELS RA100142
0.5,10/1.1,11
* QUEUES FORMED IN ACCESSING K DISKS IBM 2311 RA100020
* AND I CHANNELS RA100030
* TIME IS IN MILLISECONDS RA100040
* RA100160
* 3 VARIABLE K1(1/1) INTERVAL IN GENERATION RA100110
* 4 VARIABLE K100(1/157) ROTATION TIME RA100110
* 3 FVARIABLE K1(1/10) INTERVAL OF ARRIVALS RA
* 3 VARIABLE K1*K1 INTERVAL OF ARRIVALS RA
* 3 FVARIABLE K1(1/100) INTERVAL OF ARRIVALS RA
* 3 VARIABLE K1*K00
* 3 VARIABLE K1*K200
* THIS VARIABLE IS USED TO GENERATE TRANSACTIONS
* 3 VARIABLE K1*0.001
* 4 FVARIABLE K100(1/157) ROTATION TIME RA
* CHANGE THE ABOVE CARD FOR LONGER RECORD RA100000
7 VARIABLE 1/V3
* 7 VARIABLE GIVES THE NUMBER OF ZERO ENTRIES
6 VARIABLE K1
7 FVARIABLE (QT1+QT2+QT3+QT4+QT5+QT6+QT7+QTP+QT7)/V6
10 VARIABLE V7+QT10
* 1/L = INTER-ARRIVAL TIME
* NOTE : L= ARRIVALS PER SECOND
3 VARIABLE K1339 INTER ARRIVAL TIME MS
* VARIABLE 3 IS USED TO INDICATE THE MEAN TIME IN GENERATION
XTIME TABLE M1,0,100,100 RA100430
DISK1 EQU 1,F,0
DISK2 EQU 2,F,0
DISK3 EQU 3,F,0
DISK4 EQU 4,F,0
DISK5 EQU 5,F,0
DISK6 EQU 6,F,0
DISK7 EQU 7,F,0
DISK8 EQU 8,F,0
DISK9 EQU 9,F,0
CHAN1 EQU 10,F,0
CHAN2 EQU 11,F,0
RMULT 643,277 CHANGE THE SPEED FOR NEW RUNS
RMULT 449,201 CHANGE THE SPEED FOR NEW RUNS
RMULT 333,233 CHANGE THE SPEED FOR NEW RUNS
RMULT 439,037 CHANGE THE SPEED FOR NEW RUNS
RMULT 585,687 CHANGE THE SPEED FOR NEW RUNS

```

	RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
	RMULT	829,651	CHANGE THE SEED FOR NEW RUNS	
	RMULT	303,180	CHANGE THE SEED FOR NEW RUNS	
	RMULT	145,253	CHANGE THE SEED FOR NEW RUNS	
	RMULT	821,805	CHANGE THE SEED FOR-	
	RMULT	631,750	CHANGE THE SEED FOR NEW RUNS	
	RMULT	135,527	CHANGE THE SEED FOR NEW RUNS	
** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS				
* FOR THE RANDOM NUMBER GENERATOR ROUTINES				
* THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS				
* AND CAN BUILD CONFIDENCE INTERVALS				
* THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED				
* FROM A TABLE OF RANDOM NUMBERS				
GEN	GENERATE	V3, FN3	NORMAL	
GEN	GENERATE	V3, V3	UNIFORM	
GEN	GENERATE	V3, FN1	POISSON	
	ASSIGN	11, FN3	ASSIGN CHANNEL NUMBER	RA100190
ASS	ASSIGN	10, FN11		
QUE	ADVANCE	J, J	DUMMY BLOCK	RL100195
	CUEJE	P10	WAIT FOR DISK TO BE FREE	RA100200
	SEIZE	P10	GET THE DISK	RA100210
	DEPART	P10	DEPART FROM AVE WAITING LINE	RA100220
	ADVANCE	75, FN1		
	QUEJE	P11	WAIT FOR THE PHYSICAL CHANNEL	RA100240
	SEIZE	P11	ACCESS THE CHANNEL	RA100250
	DEPART	P11	LEAVE THE QUEUE	RA100260
	ADVANCE	13, 12	ROTATION DELAY BEFORE TRANSMISSION	RA100270
	ADVANCE	V4, FN1		
	RELEASE	P11		RA100400
	RELEASE	P10		
	TABULATE	XTIME		
SAY	SAVEVALUE	1, V10, H		
	TERMINATE	1		RA100480
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	CLEAR	XH1-XH10		
ASS	ASSIGN	10, FN12		
3	VARIABLE	K693	INTER ARRIVAL TIME MS	
6	VARIABLE	K2		
SAV	SAVEVALUE	2, V10, H		
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	CLEAR	XH1-XH10		
3	VARIABLE	K470	INTER ARRIVAL TIME MS	
ASS	ASSIGN	10, FN13		
SAV	SAVEVALUE	3, V10, H		
6	VARIABLE	K3		
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	CLEAR	XH1-XH10		
3	VARIABLE	K359	INTER ARRIVAL TIME MS	
ASS	ASSIGN	10, FN14		

(same procedure repeated for 9 disks)

	START	50, NP		
	RESET			RL100430
	START	100		RL100440
	REPORT			
	EJECT			
	SPACE	3		
	SPACE	3		
FAC	TITLE	, THE DISK UNITS	AND THE CHANNEL	
	SPACE	3		
QUE	TITLE	, THE WAITING TIME FOR EACH UNIT		
	SPACE	3		
HSAY	TITLE	, ALL WAITING I.E. TIME OVER PERFECT SERVICE		
	SPACE	3		
	SPACE	3		

TAR	TITLE	, TIME SPENT IN THE SYSTEM
	EJECT	
	EJECT	
	END	

*** APPENDIX TWO ***

PROGRAM TS50 TEST OF TIME SHARING

AUTHOR: PROSPEP M. BERNARD

```
// JOB TS50      TEST OF TIME SHARING MODELS
// JOB TS50      TIME SHARING
  ALLOC F1=CK,F2=CK
// ASSIGN SYSC0,X'131'
*   MOUNT A WORK PACK ON 131
// DLBL IJSYSC0,'SYSTEM WORK FILE NO. 0',69/032,SD
// EXTENT SYSC0,111111,1,0,0310,610
// DLBL INTER01,'SYSTEM WORK FILE NO. 0',68/032,SD
// EXTENT SYSC0,111111,1,0,0310,610
// DLBL IJSYSC1,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYSC1,111112,8,1,2920,240,7
// DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYSC1,111112,8,1,2920,240,7
// DLBL IJSYSC2,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYSC2,111112,8,1,2929,180,12
// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYSC2,111112,8,1,2929,180,12
// DLBL IJSYSC3,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYSC3,111112,8,1,2933,180,19
// DLBL REPORG1,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYSC3,111112,8,1,2933,180,19
// EXEC DARG1
```

SIMULATE

TS120Q10

```
***** NO 1
*****
***** TIME SHARING
***** BATCH PROCESSING
***** I.E. PROCESS ALL THIS CUSTOMER NO MATTER
***** HOW MUCH TIME HE NEEDS
***** NO 2
*****
***** TIME SHARING
***** ROUND ROBIN WITHOUT OVERHEAD
***** NO 3
***** TIME SHARING
***** ROUND ROBIN WITH OVERHEAD
*****
*****
***** NO 4  SHORTEST JOB  FIRST
RMULT      303,189  CHANGE THE SEED FOR NEW RUNS
RMULT      145,253  CHANGE THE SEED FOR NEW RUNS
RMULT      821,805  CHANGE THE SEED FOR-
RMULT      631,759  CHANGE THE SEED FOR NEW RUNS
RMULT      135,527  CHANGE THE SEED FOR NEW RUNS
RMULT      585,899  CHANGE THE SEED FOR NEW RUNS
RMULT      439,037  CHANGE THE SEED FOR NEW RUNS
RMULT      033,283  CHANGE THE SEED FOR NEW RUNS
RMULT      449,301  CHANGE THE SEED FOR NEW RUNS
RMULT      683,277  CHANGE THE SEED FOR NEW RUNS
```

PAGE 1

```

      RMULT      623,419      CHANGE THE SEED FOR NEW RUNS
      PMULT      829,651      CHANGE THE SEED FOR NEW RUNS
** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS
*   FOR THE RANDOM NUMBER GENERATOR ROUTINES
*   THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS
*   AND CAN BUILD CONFIDENCE INTERVALS
*
*   THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED
*   FROM A TABLE OF RANDOM NUMBERS
      1 FUNCTION  RN1,C24
0      0      .1      .104      .2      .222      .3      .355      .4      .509      .5      .69
.6      .915      .7      1.2      .75      1.38      .8      1.6      .84      1.83      .88      2.12
.9      2.3      .92      2.52      .94      2.81      .95      2.99      .96      3.2      .97      3.5
.98      3.9      .99      4.6      .995      5.3      .998      6.2      .999      7      .9997      8
      2 FUNCTION  RN2,C24      EXP * 1000
0,000/.1,104/.2,222/.3,355/.4,509/.5,690/.6,915/.7,1220
.75,1380/.8,1600/.84,1830/.88,2120/.9,2300/.92,2520/.94,2810
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200
.999,7000/.9997,8000
*   FN2 IS THE FN1 EXP TIMES 1000 FOR MORE PRECISION WHEN USING
*   IN VARIABLE STATEMENT
*   THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC
      1 VARIABLE  90      +30      AVERAGE OVERHEAD PER RECORD X NBR OF REC
      2 VARIABLE  5      NBR OF RECORDS TO BE TRANSFERRED
      2 VARIABLE  1
      1 VARIABLE  100
      1 VARIABLE  50
      3 VARIABLE  V1*V2      ACTUAL AVERAGE OVERHEAD PER QUANTUM
      3 VARIABLE  K50
      3 VARIABLE  K0
      4 VARIABLE  2*FN2
      4 VARIABLE  FN2
*   VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED      LAST ONE IS IN
      5 VARIABLE  K250
      5 VARIABLE  K1000      QUANTUM SIZE
      5 VARIABLE  K1500      QUANTUM SIZE
      5 VARIABLE  K2000      QUANTUM SIZE
      5 VARIABLE  K500
*   VARIABLE  5 INDICATES THE SIZE      OF QUANTUM
*   IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT
      6 VARIABLE  P10-V5      REDUCE REQUIRED TIME
      7 VARIABLE  V7+1
      8 VARIABLE  P9+1
      9 VARIABLE  K5000
      9 VARIABLE  K2500
      9 VARIABLE  K2000
      9 VARIABLE  K1666
      9 VARIABLE  K1250
      9 VARIABLE  K1000
      9 VARIABLE  K800
      9 VARIABLE  K1111
*   VARIABLE  9 IS 1/L INTER ARRIVAL TIME
*   VARIABLE  4 IS 1/U SERVICE TIME
*   V4 /V9 = (1/U)/(1/L) = R.      UTILIZATION OF CPU
*   K1000 / K5000 = .2      TO VARY P FOR COND. WAITING
*   K1000 / K2500 = .4
PAGE 2

```

```

* K1000 / K1660 = .6
* K1000 / K2000 = .5
* K1000 / K1250 = .8
* K1000 / K1111 = .9
* K1000 / K1000 = 1.0
10  VARIABLE P2+1
13  VARIABLE (V4/V5)+1
GEN  GENERATE V9,FN1,,,,,F
      ASSIGN  9,K1
      ASSIGN  10,V4      REQUIRED SERVICE TIME
      ASSIGN  11,P10      GET THE REQ TIME FOR SAVE
QUE  ADVANCE  0,0
FIF1 ASSIGN  2,K1      NO BREAK FO QUEUES
*   IN DIFFERENT LEVELS  ALL QUEUES IN TS RR HAVE SAME PRIORITY
      QUEUE P2
LNK  ADVANCE  0,0  TO BE
SOR  ADVANCE  0,0
      SEIZE   CPU
      DEPART  P2
TES1 TRANSFER ,ADV3
ADV  ADVANCE  V5      QUANTUM
ADV0 ADVANCE  0,0
ADV3 ADVANCE  P10      ADVANCE FOR THE R
ADV2 ADVANCE  V3      OVERHEAD IN BATCH PROCESSING
      RELEASE CPU
UNK  ADVANCE  0,0  TO BE REPLACED
ASS1 ASSIGN  10,K0
      TEST G  P10,K0,FND  IS SERVICE COMPLETED ?
      ASSIGN  9,V8
      SPLIT   1,QUE
      TERMINATE
END  ADVANCE  0,0
      TEST G  P11,V5,END1
*   WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
*   MANY QUANTA
      TABULATE XTM1
END1 ADVANCE  0,0
      ASSIGN  6,M1  PUT TIME IN PARAMETER 6
14  VARIABLE (100*P11)/P6  0/0 OF SERVICE OVER TIME IN SYSTEM
      ASSIGN  7,V14
      TABULATE XSER
      TABULATE XTIME
      TABULATE PASS
      TERMINATE 1
XTIME TABLE M1,0,200,30
XTM1  TABLE M1,0,500,30
XSER  TABLE P7,0,10,20  % OF SERVICE OVER TIME IN SYSTEM
PASS  TABLE P9,0,1,60  DISTRIBUTION OF REQUIRED NUMBER
      START  50,NP
      RESET
      START  100
      CLEAR
TES1 TEST G  P10,V5,ADV3
FIF1 ASSIGN  2,V10  FOR NON FIFO
ADV0 TRANSFER ,ADV2
ASS1 ASSIGN  10,V6
PAGE 3

```

TS120320
TS120330

```

      STAFF      50,NP
      RESET
      START      100
      CLEAR
      CLEAR
LNK  LINK      5,P10,SOR  CHANGE THE QUEING DISCIPLINE
UNK  UNLINK    5,SOR,1
TES1 TRANSFER  ,ADV3
ASS1 ASSIGN    10,KO
      START      50,NP
      RESET
      START      100
      REPORT
      EJECT
      SPACE      3
FAC  TITLE     ,UTILIZATION OF THE CPU
      SPACE      3
QUE  TITLE     ,WAITING TIME FOR THE CPU
      SPACE      3
TAB  TITLE     ,TIME SPENT IN THE SYSTEM
      SPACE      3
      EJECT
*
      JOB              TS51
*
*****
*****
***** TIME SHARING
***** COFFMAN & KLEINROCK  MODEL
*****
***** MULTILEVEL FOREGROUND BACKGROUND MODEL
*****
***** THE UNIT GOES AT THE END OF LOWER LEVEL QUEUE
***** IF SERVICE NOT COMPLETE
*****
*****
*****
*****
*****
***** TS120020
***** TS120030
***** TS120040
***** TS120050
***** TS120070
***** TS120071
***** TS120072
***** TS120073
***** TS120074
***** TS120075
***** TS120077
***** TS120078
***** TS120080
***** TS120120
*
RMULT 303,189  CHANGE THE SEED FOR NEW RUNS
RMULT 145,253  CHANGE THE SEED FOR NEW RUNS
RMULT 821,805  CHANGE THE SEED FOR-
RMULT 631,759  CHANGE THE SEED FOR NEW RUNS
RMULT 135,527  CHANGE THE SEED FOR NEW RUNS
RMULT 585,899  CHANGE THE SEED FOR NEW RUNS
RMULT 439,637  CHANGE THE SEED FOR NEW RUNS
RMULT 033,233  CHANGE THE SEED FOR NEW RUNS
RMULT 440,301  CHANGE THE SEED FOR NEW RUNS
RMULT 683,277  CHANGE THE SEED FOR NEW RUNS
RMULT 623,419  CHANGE THE SEED FOR NEW RUNS
RMULT 829,651  CHANGE THE SEED FOR NEW RUNS
** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS
* FOR THE RANDOM NUMBER GENERATOR ROUTINES
* THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS
* AND CAN BUILD CONFIDENCE INTERVALS
PAGE 4

```

*
 * THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED
 * FROM A TABLE OF RANDOM NUMBERS

1 FUNCTION RN1,C24

0	0	.1	.104	.2	.222	.3	.355	.4	.509	.5	.69
.6	.915	.7	1.2	.75	1.38	.8	1.6	.84	1.83	.88	2.12
.9	2.3	.92	2.52	.94	2.81	.95	2.99	.96	3.2	.97	3.5
.98	3.9	.99	4.6	.995	5.3	.998	6.2	.999	7	.9997	8

2 FUNCTION RN2,C24 EXP * 1000

0,000/.1,104/.2,222/.3,355/.4,509/.5,690/.6,915/.7,1220
 .75,1380/.8,1600/.84,1830/.88,2120/.9,2300/.92,2520/.94,2810
 .95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200
 .999,7000/.9997,8000

* THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC

1 VARIABLE 100
 1 VARIABLE 50
 1 VARIABLE 90 +30 AVERAGE OVERHEAD PER RECORD X NBR OF REC
 2 VARIABLE 5 NBR OF RECORDS TO BE TRANSFERRED
 2 VARIABLE K1
 3 VARIABLE V1*V2 ACTUAL AVERAGE OVERHEAD PER QUANTUM
 3 VARIABLE K50
 3 VARIABLE K0
 4 VARIABLE 2*FN2 THE REQUIRED SERVICE FOR A UNIT
 4 VARIABLE FN2

* VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED LAST ONE IS IN

5 VARIABLE K250
 5 VARIABLE K1000 QUANTUM SIZE
 5 VARIABLE K1500 QUANTUM SIZE
 5 VARIABLE K2000 QUANTUM SIZE
 5 VARIABLE K500

* VARIABLE 5 INDICATES THE SIZE OF QUANTUM

* IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT.

6 VARIABLE P10-V5 REDUCE REQUIRED TIME
 7 VARIABLE V7+1
 8 VARIABLE P9+1
 9 VARIABLE K5000
 9 VARIABLE K2500
 9 VARIABLE K2000
 9 VARIABLE K1666
 9 VARIABLE K1250
 9 VARIABLE K1000
 9 VARIABLE K800
 9 VARIABLE K1111

* VARIABLE 9 IS 1/L INTER ARRIVAL TIME
 * VARIABLE 4 IS 1/U SERVICE TIME
 * $V4/V9 = (1/U)/(1/L) = R$ UTILIZATION OF CPU
 * K1000 / K5000 = .2 TO VARY P FOR COND. WAITING
 * K1000 / K2500 = .4
 * K1000 / K1666 = .6
 * K1000 / K2000 = .5
 * K1000 / K1250 = .8
 * K1000 / K1000 = 1.0
 * K1000 / K1111 = .9

16 VARIABLE PR-1
 10 VARIABLE K101-PR TO DETERMINE WHICH QUE IT SHOULD BE
 11 VARIABLE K2*FN1-V10 REQUIRED TIME BY A UNIT LESS SERVICE TS120220

PAGE 5


```

13  VARIABLE      (V4/V5)+1
12  VARIABLE      K100-V10  NEW PRIORITY
GEN  GENPRATE      V9,FN1,,,100,,F
    ASSIGN        9,K1
    ASSIGN        10,V4      REQUIRED SERVICE TIME
    ASSIGN        11,P10     GET THE REQ TIME FOR SAVE
QUE  ADVANCE       0,0      DUMMY
FIF1 ASSIGN        2,V13     FOR FIFO ONLY
    QUEUE         P2
    SEIZE         CPU
    DEPART        P2
TES1 TRANSFER      ,ADV3
ADV  ADVANCE       V5      QUANTUM
ADV0 ADVANCE       0,0
ADV3 ADVANCE       P10     ADVANCE FOR THE R
ADV2 ADVANCE       V3      OVERHEAD IN BATCH PROCESSING
    RFLEASE       CPU
ASS1 ASSIGN        10,K0
    TEST G        P10,K0,END IS SERVICE COMPLETED ?
    ASSIGN        9,V8
    PRIORITY      V16
    SPLIT        1,QUE
    TERMINATE
END  ADVANCE       0,0
    TEST G        P11,V5,END1
*   WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
*   MANY QUANTA
    TABULATE      XTM1
    ADVANCE       0,0
    ASSIGN        6,M1     PUT TIME IN PARAMETER 6
14  VARIABLE      (100*P11)/P6  O/G OF SERVICE OVER TIME IN SYSTEM
    ASSIGN        7,V14
    TABULATE      XSER
    TABULATE      XTIME
    TABULATE      PASS
    TERMINATE     1
XTIME TABLE      M1,0,200,30
XTM1 TABLE      M1,0,500,30
XSER TABLE      P7,0,10,20  % OF SERVICE OVER TIME IN SYSTEM
PASS TABLE      P9,0,1,60  DISTRIBUTION OF REQUIRED NUMBER
    START
    RESET
    START        100
    CLEAR
FIF1 ASSIGN        2,V10     FOR NON FIFO
TES1 TEST G        P10,V5,ADV3
ADV0 TRANSFER      ,ADV2
ASS1 ASSIGN        10,V6
ADV  ADVANCE       V5      QUANTUM
ADV2 ADVANCE       V3      OVERHEAD
ASS1 ASSIGN        10,V6
    START
    RESET
    START        100
    REPORT
    EJECT

```

TS120230

TS120150

TS120320

TS120330

```

SPACE      3
FAC  TITLE  ,UTILIZATION OF THE CPU
SPACE      3
QUE  TITLE  ,WAITING TIME FOR THE CPU
SPACE      3
TAB  TITLE  ,TIME SPENT IN THE SYSTEM
SPACE      3
EJECT
END
/* END OF DATA
  ALLOC F1=CK,F2=OK
  // ASSIGN SYS(0),X'131'
  *   MOUNT A WORK PACK ON 131
  // DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD
  // EXTENT SYS000,111111,1,0,0310,610
  // DLBL INTER01,'SYSTEM WORK FILE NO. 0',69/032,SD
  // EXTENT SYS000,111111,1,0,0310,610
  // DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD
  // EXTENT SYS001,111112,8,1,2920,240,7
  // DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
  // EXTENT SYS001,111112,8,1,2920,240,7
  // DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD
  // EXTENT SYS002,111112,8,1,2929,180,12
  // DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
  // EXTENT SYS002,111112,8,1,2929,180,12
  // DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD
  // EXTENT SYS003,111112,8,1,2933,180,19
  // DLBL REPRG1,'SYSTEM WORK FILE NO. 3',67/365,SD
  // EXTENT SYS003,111112,8,1,2933,180,19
  // EXEC DARG1

```

SIMULATE

TS120010

```

*
*****
*****
***** TIME SHARING
*****
*****
***** TS120020
***** TS120030
***** TS120040

```

***** NO 4 SHORTEST JOB FIRST

TS120120

```

*
** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS
* FOR THE RANDOM NUMBER GENERATOR ROUTINES
* THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS
* AND CAN BUILD CONFIDENCE INTERVALS
*

```

```

* THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED
* FROM A TABLE OF RANDOM NUMBERS

```

```

1 FUNCTION RN1,C24
0 0 .1 .104 .2 .222 .3 .355 .4 .509 .5 .69
.6 .915 .7 1.2 .75 1.38 .8 1.6 .84 1.83 .88 2.12
.9 2.3 .92 2.52 .94 2.81 .95 2.99 .96 3.2 .97 3.5
.98 3.9 .99 4.6 .995 5.3 .998 6.2 .999 7 .9997 8
2 FUNCTION RN2,C24 EXP * 1000
0,000/.1,114/.2,222/.3,355/.4,509/.5,690/.6,915/.7,1220
.75,138/.8,160/.9,183/.94,212/.98,239/.92,252/.94,2810
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200
.999,7000/.9997,8000

```

PAGE 7

* THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC

3 FUNCTION RV1,C25 NORMAL DISTRIBUTION

0.0/0.00003,1.0/0.00135,2.0/0.00621,2.5/0.02275,3.0/0.06681,3.5
 0.11577,3.7/0.15866,4.0/0.21186,4.2/0.27425,4.4/0.34458,4.6
 0.42074,4.8/0.50000,5.0
 1.57926,5.2/0.65542,5.4/0.72575,5.6/0.78814,5.8/0.84134,6.0
 0.88493,6.2/0.93319,6.5/0.97725,7.0/0.99379,7.5/0.99865,8.0
 0.99997,9.0/1.00000,10.0

1 VARIABLE 100
 1 VARIABLE 50
 1 VARIABLE 90 +30 AVERAGE OVERHEAD PER RECORD X NBR OF REC
 2 VARIABLE 5 NBR OF RECORDS TO BE TRANSFERRED
 2 VARIABLE K1
 3 VARIABLE V1*V2 ACTUAL AVERAGE OVERHEAD PER QUANTUM
 3 VARIABLE K50
 3 VARIABLE K0
 4 VARIABLE 2*FN2 THE REQUIRED SERVICE FOR A UNIT
 4 VARIABLE FN2

* VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED LAST ONE IS IN

5 VARIABLE K250
 5 VARIABLE K1000 QUANTUM SIZE
 5 VARIABLE K1500 QUANTUM SIZE
 5 VARIABLE K2000 QUANTUM SIZE
 5 VARIABLE K500

* VARIABLE 5 INDICATES THE SIZE OF QUANTUM

* IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT

6 VARIABLE P10-V5 REDUCE REQUIRED TIME
 7 VARIABLE V7+1
 8 VARIABLE P9+1
 17 VARIABLE K160 = K800/5 FOR NORMAL
 17 VARIABLE K667 = K3333/5 FOR NORMAL
 17 VARIABLE K500 = K2500/5 FOR NORMAL
 17 VARIABLE K1000 = K5000/5 FOR NORMAL
 17 VARIABLE K222 = 51111/5 FOR NORMAL
 17 VARIABLE K250 = K1250/5 FOR NORMAL
 17 VARIABLE K286 = K1429/5 FOR NORMAL
 17 VARIABLE K333 = 11666/5 FOR NORMAL
 17 VARIABLE K700 = K1000/5 FOR NORMAL
 9 VARIABLE K5000
 9 VARIABLE K800
 9 VARIABLE K1666
 9 VARIABLE K2000
 9 VARIABLE K2500
 9 VARIABLE K3333
 9 VARIABLE K1000
 9 VARIABLE K1111
 9 VARIABLE K1250
 9 VARIABLE K1429

* VARIABLE 9 IS 1/L INTER ARRIVAL TIME
 * VARIABLE 4 IS 1/U SERVICE TIME
 * $V4/V9 = (1/U)/(1/L) = R$ UTILIZATION OF CPU
 * $K1000/K5000 = .2$ TO VARY P FOR COND. WAITING
 * $K1000/K3333 = .3$
 * $K1000/K2500 = .4$
 * $K1000/K1666 = .6$
 * $K1000/K2000 = .5$

PAGE 8

```

* K1000 / K1429 = .7
* K1000 / K1250 = .8
* K1000 / K1111 = .9
* K1000 / K1000 = 1.0
* K1000 / K800 = OVERLOADED
16 VARIABLE PR-1
10 VARIABLE K101-PR TO DETERMINE WHICH QUE IT SHOULD BE
11 VARIABLE K2*FN1-V10 REQUIRED TIME BY A UNIT LESS SERVICE TS120220
13 VARIABLE (V4/V5)+1
12 VARIABLE K100-V10 NEW PRIORITY TS120230
18 VARIABLE QT1+QT2+QT3
19 VARIABLE V18+QT4+QT5+QT6+QT7
GEN GENERATE V9,V9,,,100,,F
GEN GENERATE V17,FN3,,,100,,F
GEN GENERATE V9,FN1,,,100,,F
ASSIGN 9,K1
ASSIGN 10,V4 REQUIRED SERVICE TIME
ASSIGN 11,P10 GET THE REQ TIME FOR SAVE
QUE ADVANCE 0,0 DUMMY
FIF1 ASSIGN 2,V10 FOR NON FIFO
* IN DIFFERENT LEVELS ALL QUEUES IN TS RR HAVE SAME PRIORITY
QUEUE P2
LNK LINK 5,P10,SOR CHANGE THE QUEING DISCIPLINE
SOR ADVANCE 0,0
SEIZE CPU TS120150
DEPART P2
TES1 TRANSFER ,ADV3
ADV ADVANCE V5 QUANTUM
ADV0 TRANSFER ,ADV2
ADV3 ADVANCE P10 ADVANCE FOR THE R
ADV2 ADVANCE V3 OVERHEAD IN BATCH PROCESSING
RELEASE CPU
UNK UNLINK 5,SOR,1
ASS1 ASSIGN 10,K0
TEST G P10,K0,END IS SERVICE COMPLETED ?
ASSIGN 9,V8
PRIORITY V16
SPLIT 1,QUE
TERMINATE
END ADVANCE 0,0
TEST G P11,V5,END1
* WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
* MANY QUANTA
TABULATE XTM1
END1 ADVANCE 0,0
ASSIGN 6,M1 PUT TIME IN PARAMETER 6
14 VARIABLE (100*P11)/P6 (%) OF SERVICE OVER TIME IN SYSTEM
ASSIGN 7,V14
TABULATE XSER
TABULATE XTIME
TABULATE PASS
SAVEVALUE 1,QT1
SAVEVALUE 3,V18
SAVEVALUE 7,V19
TERMINATE 1
XTIME TABLE M1,0,200,30
PAGE 9

```

			% OF SERVICE OVER TIME IN SYSTEM DISTRIBUTION OF REQUIRED NUMBER	
XTM1	TABLE	M1,0,500,30		
XSER	TABLE	P7,0,10,20		
PASS	TABLE	P9,0,1,60		
	START	50,NP		TS120320
	RESET			TS120330
	START	100		
	RMULT	145,253	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	821,805	CHANGE THE SEED FOR-	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	631,759	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	135,527	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	585,889	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	439,037	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	033,283	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	449,301	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	683,277	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		

```

RMULT      303,199      CHANGE THE SEED FOR NEW RUNS
CLEAR
START      50,NP
RESET
START      100
REPORT
EJECT
SPACE      3
FAC  TITLE  ,UTILIZATION OF THE CPU
SPACE      3
QUE  TITLE  ,WAITING TIME FOR THE CPU
SPACE      3
TAB  TITLE  ,TIME SPENT IN THE SYSTEM
SPACE      3
SPACE      3
SAV  TITLE  ,TOTAL WAITING TIME IN QUEUE
SPACE      3
EJECT
JOB

/* END
/C  END
ALLOC F1=CK,F2=CK
// ASSIGN SYS000,X'131'
*   MOUNT A WORK PACK ON 131
// DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD
// EXTENT SYS000,111111,1,0,0310,610
// DLBL INTER01,'SYSTEM WORK FILE NO. 0',63/032,SD
// EXTENT SYS000,111111,1,0,0310,610
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYS001,111112,8,1,2920,240,7
// DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYS001,111112,8,1,2920,240,7
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYS002,111112,8,1,2929,180,12
// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYS002,111112,8,1,2929,180,12
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYS003,111112,8,1,2933,180,19
// DLBL RFPORG1,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYS003,111112,8,1,2933,180,19
// EXEC DARC1
      REALLOCATE QUE,110,XAC,100,FSV,0,MAC,0,LOG,20
      SIMULATE

*
*****
*****
***** TIME SHARING
***** II ROUND RUBIN
***** II FEED BACK FOR
***** SAME AS FB BUT A LIMITED NUMBER OF QUEUES
*****
***** MULTILEVEL FOREGROUND BACKGROUND MODEL
*****
***** THE UNIT GOFS AT THE END OF LOWER LEVEL QUEUE
***** IF SERVICE NOT COMPLETE
*****
***** TS120010
***** TS120020
***** TS120030
***** TS120040
***** TS120070
***** TS120071
***** TS120072
***** TS120073
***** TS120074
***** TS120075

```


 *

***** TS120077
 ***** TS120078
 ***** TS120080
 TS120120

RMULT	303,189	CHANGE THE SEED FOR NEW RUNS
RMULT	145,253	CHANGE THE SEED FOR NEW RUNS
RMULT	821,895	CHANGE THE SEED FOR-
RMULT	631,759	CHANGE THE SEED FOR NEW RUNS
RMULT	135,527	CHANGE THE SEED FOR NEW RUNS
RMULT	585,889	CHANGE THE SEED FOR NEW RUNS
RMULT	439,037	CHANGE THE SEED FOR NEW RUNS
RMULT	033,283	CHANGE THE SEED FOR NEW RUNS
RMULT	449,301	CHANGE THE SEED FOR NEW RUNS
RMULT	683,277	CHANGE THE SEED FOR NEW RUNS
RMULT	623,419	CHANGE THE SEED FOR NEW RUNS
RMULT	929,651	CHANGE THE SEED FOR NEW RUNS

** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS
 * FOR THE RANDOM NUMBER GENERATOR ROUTINES
 * THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS
 * AND CAN BUILD CONFIDENCE INTERVALS
 *

* THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED
 * FROM A TABLE OF RANDOM NUMBERS

1 FUNCTION		RN1,C24									
0	0	.1	.104	.2	.222	.3	.355	.4	.509	.5	.69
.6	.915	.7	1.2	.75	1.38	.8	1.6	.84	1.83	.88	2.12
.9	2.3	.92	2.52	.94	2.81	.95	2.99	.96	3.2	.97	3.5
.98	3.9	.99	4.6	.995	5.3	.998	6.2	.999	7	.9997	8
2 FUNCTION		RN2,C24 EXP * 1000									
0,000/.1,104/.2,222/.3,355/.4,509/.5,690/.6,915/.7,1220											
.75,1380/.8,1600/.84,1830/.88,2120/.9,2300/.92,2520/.94,2810											
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200											
.999,7000/.9997,8000											

* THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC

1	VARIABLE	100	
1	VARIABLE	50	
1	VARIABLE	90	+30 AVERAGE OVERHEAD PER RECORD X NBR OF REC
2	VARIABLE	5	NBR OF RECORDS TO BE TRANSFERRED
2	VARIABLE	K1	
3	VARIABLE	K50	
3	VARIABLE	10	OVERHEAD
3	VARIABLE	K0	
3	VARIABLE	K0	
3	VARIABLE	K100	SET UP TIME
4	VARIABLE	FN2	
4	VARIABLE	FN2*40	
4	VARIABLE	FN2*60	
4	VARIABLE	FN2*120	
4	VARIABLE	FN2*160	
4	VARIABLE	FN2*100	U = 0.01 1/U = 100 SEC
*	VARIABLE NO 4	IS FOR	TIME OF SERVICE NEEDED LAST ONE IS IN
5	VARIABLE	K100	QUANTUM SIZE EXCL OVERHEAD
5	VARIABLE	K200	QUANTUM SIZE EXCL OVERHEAD
5	VARIABLE	K300	QUANTUM SIZE EXCL OVERHEAD
5	VARIABLE	K400	QUANTUM SIZE EXCL OVERHEAD
5	VARIABLE	K500	

```

5    VARIABLE    K800    QUANTUM SIZE EXCL OVERHEAD
5    VARIABLE    K1000   QUANTUM SIZE EXCL OVERHEAD
5    VARIABLE    K1500   QUANTUM SIZE
5    VARIABLE    K2000   QUANTUM SIZE
5    VARIABLE    K20000
5    VARIABLE    K10000   10 SEC QUANTUM SIZE
*    IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT
*    VARIABLE    5 INDICATES THE SIZE OF QUANTUM
*    VARIABLE    5 INDICATES THE SIZE OF QUANTUM
*    IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT
6    VARIABLE    P10-V5   REDUCE REQUIRED TIME
7    VARIABLE    V7+1
8    VARIABLE    P9+1
*    VARIABLE    4 IS 1/U SERVICE TIME
*    V4 /V9 = (1/U)/1/L = R. UTILIZATION OF CPU
*    K1000 / K5000 = .2 TO VARY P FOR COND. WAITING
*    K1000 / K2500 = .4
*    K1000 / K1660 = .6
*    K1000 / K2000 = .5
*    K1000 / K1250 = .8
*    K1000 / K1000 = 1.0
*    K1000 /K1111 = .9
9    VARIABLE    K2500
9    VARIABLE    K2000
9    VARIABLE    K1666
9    VARIABLE    K1250
9    VARIABLE    K1000
9    VARIABLE    K800
9    VARIABLE    K1111
9    VARIABLE    K100000
9    VARIABLE    K5000
*    V4/V9 100000/117000 = .85
9    VARIABLE    K117747
*    VARIABLE    9 IS 1/L INTER ARRIVAL TIME
10   VARIABLE    K101-PR TO DETERMINE WHICH QUE IT SHOULD BE
11   VARIABLE    K2*FN1-V10 REQUIRED TIME BY A UNIT LESS SERVICE TS120220
12   VARIABLE    K100-V10 NEW PRIORITY TS120230
13   VARIABLE    (V4/V5)+1
14   VARIABLE    ((100*P11)/P6 C/O OF SERVICE OVER TIME IN SYSTEM
15   VARIABLE    K3 NUMBER OF USE
16   VARIABLE    PR-1
17   VARIABLE    K10
17   VARIABLE    K20
17   VARIABLE    K50
***** V7 IS THE R IE NUMBER OF QUEUES IN FOR
17   VARIABLE    K3
17   VARIABLE    K100 L
18   VARIABLE    V9/V15 1/L = 1/L
GEN  GENERATE    V9, FN1,,,100,,F
ASSIGN 9,K1
ASSIGN 10,V4 REQUIRED SERVICE TIME
ASSIGN 11,P10 GET THE REQ TIME FOR SAVE
QUE ADVANCE 0,0 DUMMY
TEST LE V10,V17,NOAS
FIF1 ASSIGN 2,V10 FOR NON FIFO
NOAS ADVANCE 0,0
PAGE 13

```


TS120150

```

        QUEUE      P2
        SE17E      CPU
        DEPART     P2
TES1  TRANSFER    ,ADV3
ADV   ADVANCE     V5          QUANTUM
ADV0  ADVANCE     C,0
ADV3  ADVANCE     P10        ADVANCE FOR THE R
ADV2  ADVANCE     V3          OVERHEAD IN BATCH PROCESSING
      RELEASE     CPU
ASS1  ASSIGN      10,KC
      TEST G      P10,K0,END  IS SERVICE COMPLETED ?
      ASSIGN      9,V8
      PRIORITY    V16
      SPLIT       1,QUE
      TERMINATE
END   ADVANCE     0,0
      TEST L      P11,K40000,END40
      TABULATE    XTM2
      TRANSFER    ,ADV66
END40 ADVANCE     0,0
      TEST L      P11,K80000,END80
      TABULATE    XTM3
      TRANSFER    ,ADV66
END80 ADVANCE     0,0
      TEST L      P11,120000,END12
      TABULATE    XTM4
      TRANSFER    ,ADV66
END12 ADVANCE     0,0
      TEST L      P11,160000,END16
      TABULATE    XTM5
      TRANSFER    ,ADV66
END16 ADVANCE     0,0
ADV66 ADVANCE     0,0
      TEST G      P11,V5,END1
*   WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
*   MANY QUANTA
      TABULATE    XTM1
END1  ADVANCE     0,0
      ASSIGN      6,M1      PUT TIME IN PARAMETER 6
      ASSIGN      7,V14
      TABULATE    XSER
      TABULATE    XTIME
      TABULATE    PASS
      TERMINATE    1
XTM1  TABLE     M1,0,50000,30
XTM2  TABLE     M1,0,50000,30
XTM3  TABLE     M1,0,50000,30
XTM4  TABLE     M1,0,50000,30
XTM5  TABLE     M1,0,50000,30
XTM1  TABLE     M1,0,50000,30
XTIME TABLE     M1,0,50000,30
XSER  TABLE     P7,C,10,20
PASS  TABLE     P9,0,1,60
      START      50,NP
      RESET
      STAPT      100

```

% OF SERVICE OVER TIME IN SYSTEM
DISTRIBUTION OF REQUIRED NUMBER

TS120320
TS120330

```

      CLEAR
TES1  TEST G      P10,V5,ADV3
ADV0  TRANSFER   ,ADV2
ASS1  ASSIGN     10,V6
ADV   ADVANCE    V5      QUANTUM
      START      50,NP
      RESET
      START      100
      REPORT
      EJECT
      SPACE      3
      SPACE      3
FAC   TITLE      ,UTILIZATION OF THE CPU
      SPACE      3
QUE   TITLE      ,WAITING TIME FOR THE CPU
      SPACE      3
TAB   TITLE      ,TIME SPENT IN THE SYSTEM
      SPACE      3
      EJECT
      END
/* END
/& END
END OF DATA
PAGE 15

```

*** A P P E N D I X T H R E E ***

PROGRAM NO RL10 - REAL TIME MODEL
AUTHOR: PROSPER M. BERNARD

// JOB RL10 REAL TIME

ALLOC F1=LK,F2=OK

// ASSIGN SYSU0,X'131'

* MOUNT A WORK PACK ON 131

// DLBL IJSYS0,'SYSTEM WORK FILE NO. 0',68/032,SD

// EXTENT SYS00,111111,1,C,0310,610

// DLBL INTER01,'SYSTEM WORK FILE NO. 0',68/032,SD

// EXTENT SYS00,111111,1,C,0310,610

// DLBL IJSYS1,'SYSTEM WORK FILE NO. 1',67/365,SD

// EXTENT SYS01,111112,8,1,2920,240,7

// DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD

// EXTENT SYS01,111112,8,1,2920,240,7

// DLBL IJSYS2,'SYSTEM WORK FILE NO. 2',67/365,SD

// EXTENT SYS02,111112,8,1,2929,180,12

// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD

// EXTENT SYS02,111112,8,1,2929,180,12

// DLBL IJSYS3,'SYSTEM WORK FILE NO. 3',67/365,SD

// EXTENT SYS03,111112,8,1,2933,180,19

// DLBL REPORG1,'SYSTEM WORK FILE NO. 3',67/365,SD

// EXTENT SYS03,111112,8,1,2933,180,19

// EXEC DARM1

SIMULATE

***** A REAL TIME INQUIRY SYSTEM *****

RL100010

RL100020

RL100030

RL100040

RL100050

RL100060

RL100090

RL100091

RL100092

RL100093

RL100150

* MEAN IS 5.0

* TIME IS IN SECONDS

XTIME TABLE M1,0,5000,20

TER1 EQU 1,F,Q

TER2 EQU 2,F,Q

PAGE 1

TER3	FQU	3, F, Q
TER4	EQU	4, F, Q
TER5	FQU	5, F, Q
TER6	EQU	6, F, Q
TER7	EQU	7, F, Q
TER8	EQU	8, F, Q
TER9	EQU	9, F, Q
TER10	EQU	10, F, Q
TER11	EQU	11, F, Q
TER12	EQU	12, F, Q
TER13	EQU	13, F, Q
TER14	EQU	14, F, Q
TER15	EQU	15, F, Q
TER16	EQU	16, F, Q
TER17	EQU	17, F, Q
TER18	EQU	18, F, Q
TER19	EQU	19, F, Q
TER20	EQU	20, F, Q
CPU	EQU	21, F, Q

* FOR MORE ACCURACY THE TIME IS IN MILLISECONDS I.E. 1/1000 OF SEC
 ** THERE ARE MANY REPLICATIONS OF THE RUN USING DIFFERENT SEEDS
 * FOR THE RANDOM NUMBER GENERATOR ROUTINES
 * THIS WAY WE OBTAIN A VARIATION OF THE SAMPLE MEANS
 * AND CAN BUILD CONFIDENCE INTERVALS
 *

* THE NUMBERS FOR THE SEEDS HAVE BEEN OBTAINED
 * FROM A TABLE OF RANDOM NUMBERS

GEN	GENERATE	1000, FN1	1 PER SEC
GEN	GENERATE	5000, FN1	.2 PER SEC
GEN	GENERATE	1250, FN1	.8 PER SEC
GEN	GENERATE	1500, FN1	.66 PER SEC.
GEN	GENERATE	1800, FN1	.55 PER SEC.
GEN	GENERATE	2222, FN1	.45 PER SEC
GEN	GENERATE	3333, FN1	.30 PER SEC
GEN	GENERATE	10000, FN1	.SEC PER SEC
GEN	GENERATE	0667, FN3	I-ARR TIME 3333 MS RATE .30 PER SEC
GEN	GENERATE	0200, FN3	I-ARR TIME 1000 MS RATE 1.0 PER SEC
GEN	GENERATE	0250, FN3	I-ARR TIME 1250 MS RATE .80 PER SEC
GEN	GENERATE	0300, FN3	I-ARR TIME 1500 MS RATE .66 PER SEC
GEN	GENERATE	0360, FN3	I-ARR TIME 1800 MS RATE .55 PER SEC
GEN	GENERATE	0400, FN3	I-ARR TIME 2000 MS RATE .50 PER SEC
GEN	GENERATE	0500, FN3	I-ARR TIME 2500 MS RATE .40 PER SEC
GEN	GENERATE	1000, FN3	I-ARR TIME 5000 MS RATE .20 PER SEC
GEN	GENERATE	2000, FN3	I-ARR TIME 10000 MS RATE .10 PER SEC
GEN	GENERATE	1000, 1000	UNIFORM DISTRIBUTION
GEN	GENERATE	1250, 1250	UNIFORM DISTRIBUTION
GEN	GENERATE	1500, 1500	UNIFORM DISTRIBUTION
GEN	GENERATE	1800, 1800	UNIFORM DISTRIBUTION
GEN	GENERATE	2000, 2000	UNIFORM DISTRIBUTION
GEN	GENERATE	2222, 2222	UNIFORM DISTRIBUTION
GEN	GENERATE	5000, 5000	UNIFORM DISTRIBUTION
GEN	GENERATE	10000, 10000	UNIFORM DISTRIBUTION
GEN	GENERATE	3333, 3333	UNIFORM DISTRIBUTION
GEN	GENERATE	0444, FN3	I-ARR TIME 2222 MS RATE .45 PER SEC
GEN	GENERATE	0666, FN3	I-ARR TIME 3333 MS RATE .35 PER SEC
GEN	GENERATE	2500, 2500	UNIFORM DISTRIBUTION

GEN	GENERATE	2500, FN1	.40 PER SEC.	
GEN	GENERATE	2500, FN1	.50 PER SEC	
	ASSIGN	12, FN2	TERMINAL NUMBER	RL100177
	QUEUE	P12	WAIT TERMINAL	RL100187
	SEIZE	P12	GET TERMINAL	RL100197
	DEPART	P12	DEPART FROM QUEUE	RL100200
			KEY IN TIME IS	RL100210
			10 + OR - 5 SECONDS	RL100227
	ADVANCE	10000, 5000		
	QUEUE	21 CPU		
	SEIZE	21 CPU		
	DEPART	21 CPU		
	ADVANCE	1000, FN1	APPROXIMATION OF ERLANG -2 IS TWO EXP	
	ADVANCE	1000, FN1	APPROXIMATION OF ERLANG -2 IS TWO EXP	
			CPU TIME	RL100300
	RELEASE	21 CPU		
	ADVANCE	5000, 2000		
	RELEASE	P12		RL100240
	TABULATE	XTIME		RL100400
	TERMINATE	1		RL100410
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	145, 253	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	821, 805	CHANGE THE SEED FOR-	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	631, 759	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	135, 527	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	585, 889	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	439, 037	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	033, 283	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50, NP		RL100430
	RESET			RL100440

START	100		
RMULT	449,301	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RFSET			RL100440
START	100		
RMULT	683,277	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESFT			RL100440
START	100		
RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	829,651	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESFT			RL100440
START	100		
REPORT			
EJECT			
SPACE	3		
FAC	TITLE	,TERMINAL UTILIZATION	
	SPACE	3	
QUE	TITLE	,THE WAITING FOR THE TERMINALS	
	EJECT		
TAB	TITLE	,THE TRANSIT TIME IN THE SYSTEM	
	SPACE	3	
	EJECT		
	END		RL100460
/* END OF DATA			
/E END OF JOB			
END OF DATA			
PAGE 4			

*** APPENDIX FOUR ***

Program CONLM1 - confidence limits

LIST
CONLM1

```

9000 REM CONLM1--COMPUTES CONFIDENCE LIMITS FOR UNKNOWN POPULATION MEAN
9002 DATA 5.E+06,5.39825E+06,5.7926E+06,6.17911E+06,6.55422E+06,6.9146E
9004 DATA 7.58036E+06,7.88145E+06,8.1594E+06,8.41345E+06,8.64334E+06,8.8
9006 DATA 9.19243E+06,9.33193E+06,9.45201E+06,9.55435E+06,9.6407E+06,9.7
9008 DATA 9.82136E+06,9.86097E+06,9.89276E+06,9.91803E+06,9.9379E+06,9.9
9010 DATA 9.97445E+06,9.98134E+06,9.98652E+06,9.99032E+06,9.99313E+06,9.9
9012 DATA 9.99767E+06,9.99841E+06,9.99899E+06,9.99928E+06,9.99952E+06,9.9
9014 DATA 9.99967E+06,9.99992E+06,9.99995E+06,9.99997E+06,9.99998E+06,9.9
9016 DIM X(49)
9018 DEF FND(V)=M+V*S
9020 DEF FND(V)=X(V+1)-X(V)
9022 DEF FND(V)=U-V*(V-1)*(D2/(2*D1)+(V-2)*D3/(6*D1))
9024 DEF FNZ(V)=1+(V+2)/(4*D)+((V+3)*(5+V+2))/(96*D+2)
9026 DATA 0,0,0
9028 FOR I=1 TO 49
9030 READ X(I)
9032 NEXT I
9034 READ I,S1,S2
9036 READ H
9038 LET I=I+1
9040 READ Q
9042 IF V=3.E+31 THEN 9050
9044 LET S1=S1+V
9046 LET S2=S2+V^2
9048 GOTO 9038
9050 LET N=I-1
9052 PRINT
9054 PRINT "VALUES OF SAMPLE STATISTICS:"
9056 PRINT
9058 PRINT "    SIZE OF SAMPLE",N
9060 LET M=S1/N
9062 PRINT "    SAMPLE MEAN VALUE",M
9064 LET S8=S2/N-M^2
9066 LET D=N-1
9068 PRINT "    VARIANCE OF SAMPLE",S8
9070 PRINT "    SAMPLE STD DEVIATION",SOR(S8)
9072 LET S5=S2+N*(H-1)/(H*(N-1))
9074 PRINT "    ESTIMATED POPN STD DEV",SOR(S5)
9076 LET S6=S5*(H-N)/(N*(H-1))
9078 LET S=SOR(S6)
9080 PRINT "    STANDARD ERROR OF MEAN",S
9082 PRINT
9084 PRINT
9086 PRINT "CONFIDENCE LIMITS ON POPULATION MEAN:"
9088 PRINT
9090 PRINT "CONF LEVEL","LOWER LIM","UPPER LIM"
9092 PRINT
9094 READ P
9096 IF P=1.E+32 THEN 9999
9098 LET A1=.5*(1+P)
9100 GOSUB 9106
9102 PRINT 100+P,FND(-A2+FNZ(A2)),FND(A2+FNZ(A2))
9104 GOTO 9094
9106 IF A1>.5 THEN 9116
9108 LET A1=1-A1
9110 GOSUB 9124
9112 LET A2=-Q
9114 GOTO 9120
9116 GOSUB 9124
9118 LET A2=Q
9120 RETURN
9122 REM REVERSE INTERPOLATION FOR STD NORMAL DEVIATE:
9124 LET Z=1.E+07*A1
9126 FOR I=1 TO 49
9128 IF Z<X(I) THEN 9134
9130 NEXT I
9132 LET I=I-1
9134 LET I=I-1
9136 LET D1=FND(I)
9138 LET E1=FND(I+1)
9140 LET D2=E1-D1
9142 LET D3=FND(I+2)-E1-D2
9144 LET U=(Z-X(I))/D1
9146 LET Q=FND(FND(U))
9148 LET Q=.1*(Q+1-1)
9150 RETURN
9900 DATA 1.E+20
9901 DATA 65.4,106.43,7.47,2
9902 DATA 95.8,71.1,6.0,2,57.2
9903 DATA 71.2,74.9,6.7,71.5
9904 DATA 3.E+31,.5,.75,.9,.95,.99,.999,.9999,.99999,1.E+32
9999 END

```

*** A P P E N D I X F I V E ***

PROGRAM FINADAV - ANALYSIS OF VARIANCE
 AUTHORS: WRITTEN BY LAWRENCE JORDAN
 MODIFIED BY PROSPER M. BERNARD

```
// JOB FINADAV      ANALYSIS OF VARIANCE
// CPTION CATAL,NODECK
// PHASE FINADAV2,*
// EXEC FFORTRAN
      DIMENSION A(20), B(20), AB(20,20), SA(20), X(200), DF(5), EFF(5),
      + TIT(40), FMT(72)

C
C   SPLIT-PLOT DESIGN (WINER, 1971, PP. 518-539, 599-603)
C
C   CONTROL VARIABLES--IN 5-DIGIT FIELDS
C
C   P = LEVELS OF A (BETWEEN SUBJECTS FACTORS)
C   Q = LEVELS OF B (WITHIN SUBJECTS FACTORS)
C   S = NUMBER OF SUBJECTS OR 'PLOTS'
C   NVAR = NUMBER OF DEPENDENT VARIABLES (USUALLY 1)
C
C   NEXT CARD READ IS A FORMAT SPECIFICATION, HANDLED BY FMTS
C   SUBROUTINE.  EACH READING OF THE FORMAT CARD MUST SPECIFY--
C       LEVEL OF A
C       NVAR OBSERVATIONS IN CELL B(1)
C       NVAR OBSERVATIONS IN CELL B(2)
C       AND SO ON, UP TO
C       NVAR OBSERVATIONS IN CELL B(Q)
C
C   FOLLOWING THE INPUT DATA, WHICH IS PRINTED OUT, THE PROGRAM EXPECTS
C   ONE TITLE CARD FOR EACH OF THE NVAR DEPENDENT VARIABLES
C
      IN = 8
      IOUT = 5
      WRITE (IOUT,1)
1    FORMAT(1H1)
      READ ( IN, 1000 )  P, Q, S, NVAR
1000  FORMAT(3F5.0,I5 )
      IP = P
      IQ = Q
      IS = S
      DF(1) = P - 1
      DF(2) = S - P
      DF(3) = Q - 1
      DF(4) = (P - 1)*(Q - 1)
      DF(5) = (S - P)*(Q - 1)
      WRITE(IOUT,111) IP, IQ, IS, NVAR
111  FORMAT(' LEVELS OF A =',I5/' LEVELS OF B =',I5/' NUMBER OF SUBJECT
      +S =',I5/' NUMBER OF DEPENDENT VARIABLES =',I5 )
      NFILE = NVAR * 10
      READ ( IN, 6 )  (FMT(J), J = 1, 72 )
6    FORMAT(72A1)
PAGE 1
```



```

WRITE(IOUT,112) (FMT(J), J = 1, 72)
112 FORMAT('CFORMAT SPECIFICATION = ',72A1)
WRITE(IOUT, 115)
115 FORMAT('OF TESTS ASSUME THAT SUBJECTS OR PLOTS IS A RANDOM FACTOR
+AND A AND B ARE FIXED FACTORS--CF. WINER, P. 519.')
WRITE(IOUT,114)
114 FORMAT('C INPUT DATA--')
IFILE = 1
DO 20 K = 1, IS
IF (K - 1) 7, 7, 8
C THE STATEMENT 7 HAS BEEN CHANGED FOPMAT CARD ON DATA DUMMY
C CALL FMTS(FMT,72) CHANGED BY PB 1.12.72
7 CONTINUE
8 READ (IN, 9) LA, (X(J), J = 1, NFILE )
9 FORMAT(12,10F5.0)
C FORMAT(12F6.0)
C STATEMTN 9 CNANGED PB
WRITE(IOUT,113) LA, (X(J), J = 1, NFILE)
113 FORMAT(' ',13,25F8.1)
WRITE(4) LA,(X(J), J = 1, NFILE )
20 CONTINUE
END FILE 4
REWIND 4
DO 90 K = 1, NVAR
IFILE = 1
K1 = K + (IQ - 1) * NVAR
DO 10 I = 1, IP
A(I) = 0
SA(I) = 0
DO 10 J = 1, IQ
B(J) = 0
10 AB(I,J) = 0
DO 30 I = 1, IS
30 SA(I) = 0
G = 0
H = 0
P2 = 0
SSA = 0
SSB = 0
SSAB = 0
X2 = 0
WRITE (IOUT,1)
READ (IN,3) ( TIT(L), L = 1, 40 )
3 FORMAT(40A2)
4 FORMAT(' ',40A2)
WRITE (IOUT, 4 ) ( TIT(L), L = 1, 40 )
DO 100 I = 1, IS
READ (4) LA,(X(J), J = 1, NFILE )
SA(LA) = SA(LA) + 1
L = 0
P1 = 0
DO 90 J = K, K1, NVAR
L = L + 1
X2 = X2 + X(J)*X(J)
G = G + X(J)
A(LA) = A(LA) + X(J)

```

```

      B(L) = B(L) + X(J)
      AB(LA,L) = AB(LA,L) + X(J)
90    P1 = P1 + X(J)
      P2 = P2 + P1*P1
100   CONTINUE
      P2 = P2/Q
      G2 = G*G / S / Q
      G = G / S / Q
      DO 200 I = 1, IP
      H = H + 1 / SA(I)
      SSA = SSA + A(I)*A(I)/SA(I)/Q
      DO 200 J = 1, IQ
      SSAB = SSAB + AB(I,J)*AB(I,J)/SA(I)
200   AB(I,J) = AB(I,J) / SA(I)
      H = P / H
      DO 300 J = 1, IQ
300   SSB = SSB + B(J)*B(J)/S
      EFF(1) = SSA - G2
      EFF(2) = P2 - SSA
      EFF(3) = SSB - G2
      EFF(4) = SSAB - SSA - SSB + G2
      EFF(5) = X2 - SSAB - P2 + SSA
      CALL TABLE ( EFF, DF, 1, G )
      SSA = 0
      SSB = 0
      SSAB = 0
      G = 0
      DO 500 I = 1, IQ
500   B(I) = 0
      DO 600 I = 1, IP
      A(I) = 0
      DO 600 J = 1, IQ
      A(I) = A(I) + AB(I,J)
      B(J) = B(J) + AB(I,J)
      SSAB = SSAB + AB(I,J)*AB(I,J)
600   G = G + AB(I,J)
      DO 650 I = 1, IP
650   SSA = SSA + A(I)*A(I)
      SSA = SSA/Q
      DO 700 I = 1, IQ
700   SSB = SSB + B(I)*B(I)
      SSB = SSB/P
      G2 = G*G/P/Q
      G = G / P / Q
      EFF(1) = H * (SSA - G2)
      EFF(3) = H * (SSB - G2)
      EFF(4) = H * (SSAB - SSA - SSB + G2 )
      CALL TABLE ( EFF, DF, 2, G )
      WRITE(5, 1003)
1003  FORMAT('CELL MEANS--ROWS = LEVELS OF A, COLUMNS = LEVELS OF B')
      CALL LPRIN ( AB, IP, IQ, 20 )
900   CONTINUE
      DO 920 IAB = 1, 6
      READ (IN,3) ( TIT(L), L = 1, 40 )
      WRITE (OUT,29) ( TIT(L), L = 1, 40 )
29    FORMAT(40A2)
PAGE 3

```

```

920  CONTINUE
      WRITE(IOUT,1)
      CALL EXIT
      END
      SUBROUTINE TABLE (EFF, DF, KIND, G )
      DIMENSION EFF(5), DF(5)
      IOUT = 5
      IF (KIND -1) 10, 10, 30
10    WRITE(IOUT, 1)
1    FORMAT('O  ANOVA--LEAST SQUARES SOLUTION')
      GO TO 50
30    WRITE(IOUT,2)
2    FORMAT('O  ANOV--UNWEIGHTED SOLUTION')
50    WRITE(IOUT,4)
4    FORMAT ('CSOURCE          SS          DF          MS
+      F'/O          ')
C    BETWEEN SUBJECT REMOVED PB
      SME = EFF(2)/DF(2)
      SM = EFF(1)/DF(1)
      F = SM /SME
      WRITE(5,5) EFF(1), DF(1), SM, F
5    FORMAT('O          A',F15.3,F15.0,2F15.3)
      WRITE (IOUT,6) EFF(2), DF(2), SME
6    FORMAT('O  S(A)',F15.3, F15.0, F15.3 )
      SME = EFF(5) / DF (5)
      SM = EFF(3) / DF(3)
      F = SM / SME
      WRITE(5,7) EFF(3), DF(3), SM, F
C    FORMAT('O WITHIN SUBJECTS'///'          B', F15.3, F15.0, 2F15.3 )
C    ABOVE CHANGED PB
7    FORMAT('O          '///'          B', F15.3, F15.0, 2F15.3 )
      SM = EFF(4) / DF(4)
      F = SM / SME
      WRITE(IOUT,8) EFF(4), DF(4), SM, F
8    FORMAT('          AB', F15.3, F15.0, 2F15.3)
      WRITE(IOUT,9) EFF(5), DF(5), SME
9    FORMAT('          BS(A)', F15.3, F15.0, F15.3 )
      WRITE (IOUT, 11) G
11   FORMAT('O          GRAND MEAN = ',F10.3, '// )
      RETURN
      END
      SUBROUTINE LPRIN ( A, M, N, MDIM )
      DIMENSION A(1), OUT(10), ICOL(10)

C
C  PRINTS MATRICES IN 10-COLUMN SEGMENTS AND F10.3 FORMAT
C
C  IF A IS A GENERAL (DOUBLE-DIMENSIONED) MATRIX--
C      M = NO. ROWS OF A
C      N = NO. COLS OF A
C      MDIM = ROW (FIRST) DIMENSION OF A IN CALLING PROGRAM DIMENSION
C      STATEMENT
C
C  IF A IS AN UPPER TRIANGULAR (SYMMETRIC--STORAGE MODE = 1) MATRIX--
C      M = N = NO. OF ROWS AND COLS IN OUTPUT MATRIX
C      MDIM = 0
C
PAGE 4

```

```

      IF ( MDIM ) 30, 10, 30
10    IF ( M - N ) 20, 30, 20
20    WRITE (IOUT, 4)
4     FORMAT('ATTEMPT TO PRINT SYMMETRIC MATRIX WITH M .NE. N ENCOUNTER
      +ED IN LPRIN SUBROUTINE. ')
      GO TO 900
30    IFLAG = 0
      IOUT = 5
      J1 = -9
      J2 = 0
      J5 = 10
      JSEC = 0
100   J1 = J1 + 10
      J2 = J2 + 10
      JSEC = JSEC + 1
      IF ( J2 - N ) 110, 110, 115
110   WRITE (IOUT, 1) JSEC
1     FORMAT (11H0 SECTION 13 / )
115   IF ( J2 - N ) 130, 120, 120
120   IFLAG = 1
      J2 = N
130   CONTINUE
      J5 = 0
      DO 140 I = J1, J2
      J5 = J5 + 1
140   ICOL( J5 ) = I
      WRITE (IOUT, 2) ( ICOL( I ), I = 1, J5 )
2     FORMAT('0', 6X, 10( 9X, 12 ) / )
      DO 200 I = 1, M
      J3 = 0
      DO 150 J = J1, J2
      IF ( MDIM ) 142, 144, 142
142   LOC = J * MDIM - MDIM + 1
      GO TO 149
144   IF (J-J1) 145, 145, 146
145   LOC = (J1 + I - 1)*(J1 + I - 2)/2 - (JSEC - 1)*10
146   IF (J-I) 147, 147, 148
147   LOC = LOC + 1
      GO TO 149
148   LOC = LOC + J - 1

149   J3 = J3 + 1
150   OUT( J3 ) = A( LOC )
      WRITE (IOUT, 3) I, (OUT(J4), J4 = 1, J3)
3     FORMAT(' ', 1X, 15, 4X, 10( F10.3, 1X ) )
200   CONTINUE
      IF ( IFLAG - 1 ) 100, 900, 100
900   RETURN
      END
/* END
// EXEC LNKEOT
/& END
END OF DATA
PAGE 5

```

IBM 2311. DISK UNIT

STORAGE CAPACITY: 7.25 million bytes per disk pack
36,000 bytes per cylinder

ACCESS TIME: Minimum is 25 ms
Maximum is 135 ms
Average is 74.25 ms

ROTATION TIME: At 2400 revolutions per minute, rotation
time is 25 ms
On average rotational delay is 12.5 ms

TRANSMISSION TIME: 156 thousand bytes per second
0.0064103 milliseconds per bytes on
average

RECORD LENGTH: In our application record length is 100 bytes

DATA REMOVABILITY: To physically change a disk pack it takes
one minute.

TABLE 2.1: Timing on IBM 2311 Disk

Source: IBM 360 System Summary, (New York: IBM, ref: A 26-5988-6,
1969), p. 34.

*** APPENDIX SEVEN ***

PROGRAM FINCONC1 - CONFIDENCE LIMITS
AUTHOR: PROSPER M. BERNARD

```
// JOB FINCONC1  COMPILATION
// OPTION NODECK
// OPTION CATAL
// PHASE FINCONC1,*
// EXEC PL/I
* PROCESS STMT
```

```
/*  AUTHOR : PROSPER M. BERNARD  */
```

```
FINCON:
  PROCEDURE OPTIONS (MAIN);
```

```
/*  THIS PROGRAM CALCULATES THE CONFIDENCE INTERVALS  */
```

```
/*
```

```
  THIS PROGRAM CALCULATES THE CONFIDENCE INTERVALS
  FOR A SAMPLE .
```

```
  THE DATA IS ENTERED AS FOLLOWS:
```

```
FINCONC1
  CARD 1          CC1 - CC3      THE SIZE OF THE SAMPLE
  ALL OTHER CARDS INPUT DATA    4 DIGITS PER VALUE ONE DECIMAL
                                  POSITION NO DECIMAL POINT.
```

```
FINCONC2
  6 POSITION PER VALUE
```

```
FINCONC3
  5 POS 3 DEC
```

```
*/
```

```
DCL  UPDATE ENTRY,
      DATE   CHAR (8),
      UPSI   FIXED BINARY (31);
```

```
DCL  CARD1 FILE RECORD INPUT BUFFERED
      ENV (MEDIUM (SYSIPT,1442) F(90));
```

```
DCL  1 CARD1 EXT,
      2 CC1 CHAR (8);
```

```
DCL  DAT (20) PICTURE '99V999' DEFINED CARD1;
```

```
DCL  CH (80) CHAR (1) DEFINED CARD1;
```

```
DCL  X(40) FLOAT DEC (12)
```

```
  INITIAL
```

```
PAGE 1
```

```
00020
00040
00060
00080
00100
00120
00140
00160
00180
00200
00220
00240
00260
00280
00300
00320
00340
00360
00380
00400
00420
00440
00460
00480
00500

00520
00540
00560
00580
00600
00620
00640
00660
00680
00700
00720
00740
00760
00780
00800
00820
00840
```

```

        (5.E+06,5.39828E+06,5.7926F+06,6.17911E+06,
        6.55422E+06,6.91462E+06,7.25747E+06,7.59036E+06,
        7.88145E+06,8.1574E+06,8.41345E+06,8.64334E+06,
        8.8493E+06,9.132E+06,9.19243E+06,9.33193E+06,
        9.45201E+06,9.55435E+06,9.6407E+06,9.71283E+06,
        9.7725E+06,9.82136E+06,9.86097E+06,9.89276E+06,
        9.91803E+06,9.9379E+06,9.95337E+06,9.96533E+06,
        9.97445E+06,9.98134E+06,9.9865E+06,9.99032E+06,
        9.99313E+06,9.99517E+06,9.99663E+06,9.99767E+06,
        9.99841E+06,9.99892E+06,9.99928E+06,9.99952E+06,
        9.99968E+06,9.99979E+06,9.99987E+06,9.99992E+06,
        9.99995E+06,
        9.99997E+06,9.99998E+06,9.99999E+06,9.99999E+06);
DCL P (C8) FIXED DEC (15,5)
INITIAL
        (.5,.75,.9,.95,.99,.999,.9999,.99999);
DCL W(10) FLOAT DEC (12),
      F(10) FIXED DEC (12);

DCL HDATE CHAR (8) INIT ('0J/0J/00');
DCL
J  FLOAT DEC (12),
M  FLOAT DEC (12);

ON ENDFILE (CARDI) GO TO OUT;
OPEN FILE (CARDI);
READ FILE (CARDI) INTO (CARD1);
CALL ZERO;
      CALL UPDATE (UPSI,DATE);
      HDATE = DATE;
N = DAT (1);
LOOP:
      PUT EDIT (' INFERENCE FROM A SAMPLE ',HDATE)
        (PAGE,A,A(8));
I= 0;
S1 = 0;
S2 = 0;
Q = 0;
READ FILE (CARDI) INTO (CARD1);
CALL ZERO;
DO I = 1 TO N;
      W(I) = DAT (I);
END;
      H= 1.E+20 ;
      DO I= 1 TO N;
        S1 = S1 +W(I);
        S2 = S2 +W(I)**2;
      END;
      ASO:
      N = I -1;
      PUT EDIT ('VALUE OF SAMPLE STATISTICS')
        (SKIP (2),A);
      PUT EDIT ('LIST OF THE INPUT DATA') (SKIP(2),A);

```

```

00860
00890
00900
00920
00940
00960
00980
01000
01020
01040
01060
01080
01100
01120
01140
01160
01180
01200
01220
01240
01260
01280
01300
01320
01340
01360
01380
01400
01420
01440
01460
01480
01500
01520
01540
01560
01580
01600
01620
01640
01660
01680
01700
01720
01740
01760
01780
01800
01820
01840
01860
01880
01900
01920
01940
01960

```

DO IA = 1 TO (N/6);	C1980
PUT SKIP;	C2000
DO I = (6 * (IA-1) + 1) TO (6 * IA);	C2020
PUT EDIT	C2040
(DAT(I)) (F(7,3));	C2060
END;	C2100
END;	C2120
	C2140
	C2160
	C2180
	C2200
PUT EDIT ('SIZE OF THE SAMPLE : ',N)	C2220
(SKIP (1),A,F(5));	C2240
M = S1/N;	C2260
PUT EDIT ('SAMPLE MEAN VALUE : ',M)	C2280
(SKIP (1),A,F(9,4));	C2300
S8 = S2/N-M**2;	C2320
D = N-1;	C2340
PUT EDIT ('VARIANCE OF SAMPLE : ',S8) (SKIP (1),A,F(11,4));	C2360
S88 = SORT (S8);	C2380
PUT EDIT (' SAMPLE STD DEVIATION: ',S88) (SKIP (1),A,F(9,4));	C2400
S5 = S8*N*(H-1)/(H*(N-1));	C2420
S55 = SORT(S5);	C2440
PUT EDIT (' EST POP STD DEV : ',S55) (SKIP(1),A,F(9,4));	C2460
S6 = S5*(H-N)/(N*(H-1));	C2480
S = SORT (S6) ;	C2500
PUT EDIT	C2520
(' STANDARD ERROR MEANS ',S) (SKIP (1),A,F(9,4));	C2540
PUT EDIT (' CONFIDENCE LIMITS ON POPULATION MEAN: ')(SKIP (3),A);	C2560
PUT EDIT	C2580
('CONF LEVEL LOWER LIM UPPER LIM')	C2600
(SKIP(2),A);	C2620
	C2640
	C2660
	C2680
DO L = 1 TO 8 ;	C2700
A1 = .5 * (1 + P(L));	C2720
CALL A9106;	C2740
CALL PRINT;	C2760
END;	C2780
GO TO LOOP;	C2800
	C2820
	C2840
A9106:	C2860
PROCEDURE;	C2880
IF A1 > .5 THEN GO TO A9116;	C2900
A1 = 1-A1;	C2920
CALL A9124;	C2940
A2 = -Q; GO TO A9120;	C2960
A9116:	C2980
CALL A9124;	C3000
A2 = Q;	C3020
A9120:	C3040
END;	C3060
A9124:	C3080
PROCEDURE;	


```

      Z = 1.E+07*A1;
      DO I = 1 TO 49;
      IF Z < X(I) THEN GO TO A9134;
      END;
A9134:
      I=I-1;
      I=I-1;
      D1 = X(I+1) - X(I);
      E1 = X(I+2) - X(I+1);
      D2 = E1 - D1;
      D3 = X(I+3) - X(I+2);
      D3 = D3 - E1 - D2;
      U = (Z-X(I))/D1;
      J = I + 2;
      J = U;
      QZ = FNB (U,J,D2,D1,D3);
      J = QZ;
      Q = FNB (U,J,D2,D1,D3);
      Q = .1 *(Q+I-1);
      END;
/* FOLLOWING ARE ALL FUNCTIONS
   AND SUBROUTINES USED */

FNB:  PROCEDURE (U,J,D2,D1,D3);
      DCL J FLOAT DEC (12);
      WW = U-J*(J-1)*(D2/(2*D1)+(J-2)*D3/(6*D1));
      RETURN (WW);
      END;

FNZ:  PROCEDURE (A2,D)
      ;
      WWW = 1+ (A2**2+1)/(4*D)+((A2**2+3)*(5*A2**2+1))/( 96 *D**2);
      RETURN (WWW);
      END;

PRINT:  PROCEDURE;
      AF = FNZ(A2,D);
      AZ = -A2 * AF;
      AZZ = A2 * AF;
      AR = M+AZ*S;
      AL = M+AZZ*S;
      P100 = 100 * P(L);
      PUT EDIT (P100,AR,AL) (SKIP (1),F(12,6),F(14,6),F(14,6));
      END;

```

```

03100
03120
03140
03160
03180
03200
03220
03240
03260
03280
03300
03320
03340
03360
03380
03400
03420
03440
03460
03480
03500
03520
03540
03560
03580
03600
03620
03640
03660
03680
03700
03720
03740
03760
03780
03800
03820
03840
03860
03880
03900
03920
03940
03960
03980
04000
04020
04040
04060
04080
04100
04120
04140
04160
04180
04200

```

```

ZERO:
PROCEDURE:
DO IZ = 1 TO 80;
  IF
    CH (IZ) = ' ' THEN CH (IZ) = '0';
END:
RETURN:
END:

```

```

OUT:
  PUT EDIT (' END OF RUN ') (SKIP(1),A);
END:

```

```

/* END
// EXEC LNKEDE
/G END
END OF DATA
PAGE 5

```

```

04220
04240
04260
04280
04300
04320
04340
04360
04380
04400
04420
04440
04460
04480
04500
04520
04540
04560
04580
04600

```

*** APPENDIX EIGHT ***

PROGRAM PRI0 TEST OF PRIORITY MODEL
AUTHOR: PROSPER M. BERNARD

```
// JOB PRI0 PRIORITY MODEL
  ALLOC F1=OK,F2=OK
  // ASSIGN SYS000,X'131'
  * MOUNT A WORK PACK ON 131
  // DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD
  // EXTENT SYS000,111111,1,0,0310,610
  // DLBL INTER01,'SYSTEM WORK FILE NO. 0',68/032,SD
  // EXTENT SYS000,111111,1,0,0310,610
  // DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD
  // EXTENT SYS001,111112,8,1,2920,240,7
  // DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
  // EXTENT SYS001,111112,8,1,2920,240,7
  // DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD
  // EXTENT SYS002,111112,8,1,2929,180,12
  // DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
  // EXTENT SYS002,111112,8,1,2929,180,12
  // DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD
  // EXTENT SYS003,111112,8,1,2933,180,19
  // DLBL REPORG1,'SYSTEM WORK FILE NO. 3',67/365,SD
  // EXTENT SYS003,111112,8,1,2933,180,19
  // EXEC DARC1
    SIMULATE
***** NO 1
*****
***** TIME SHARING *****
*** 1.1 NON-PREEMPTIVE ONLY ONE QUEUE
*** 1.2 NON-PREEMPTIVE ONE QUEUE PER PRIORITY
*** 1.3 NON-PREEMPTIVE LIMITED QUANTUM
*** A LIMITED QUANTUM IMPLIES A ROUND ROBIN MODEL IN EACH OF THE
*** QUEUE
*** THE PRIORITIES ARE NOT CHANGED AFTER EACH PASS
***
*** 2.1 PREEMPTIVE ONLY ONE QUEUE RECORDED
*** 2.2 PREEMPTIVE ONE QUEUE FOR EACH PRIORITY
*** 2.3 PREEMPTIVE WITH LIMITED QUANTUM
*****
1 FUNCTION RN1,C24
0 .0 .1 .104 .2 .222 .3 .355 .4 .509 .5 .69
.6 .915 .7 1.2 .75 1.38 .8 1.6 .84 1.83 .88 2.12
.9 2.3 .92 2.52 .94 2.81 .95 2.99 .96 3.2 .97 3.5
.98 3.9 .99 4.6 .995 5.3 .998 6.2 .999 7 .9997 8
2 FUNCTION RN2,C24 EXP * 1000
0,000/.1,104/.2,222/.3,355/.4,509/.5,691/.6,915/.7,1220
.75,1390/.8,1600/.84,1830/.88,2120/.9,2300/.92,2520/.94,2810
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200
.999,7000/.9997,8000
* FN2 IS THE FN1 EXP TIMES 1000 FOR MORE PRECISION WHEN USING
* IN VARIABLE STATEMENT
* THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC
PAGE 1
```

TS120010

```

3      FUNCTION      RN1,C25      NORMAL DISTRIBUTION
0.0.C/0.0003,1.7/0.00135,2.0/0.00621,2.5/0.02275,3.0/0.06681,3.5
0.11507,3.7/0.15865,4.0/0.21186,4.2/0.27425,4.4/0.34458,4.6
0.42074,4.8/0.50000,5.0
0.57926,5.2/0.65542,5.4/0.72575,5.6/0.78814,5.8/0.84134,6.0
0.88492,6.2/0.93319,6.5/0.97725,7.0/0.99379,7.5/0.99865,8.0
0.99997,9.0/1.00000,10.0
1      VARIABLE      90      +30      AVERAGE OVERHEAD PER RECORD X NBR OF REC
2      VARIABLE      5      NBR OF RECORDS TO BE TRANSFERRED
2      VARIABLE      1
1      VARIABLE      100
1      VARIABLE      50
3      VARIABLE      V1*V2      ACTUAL AVERAGE OVERHEAD PER QUANTUM
3      VARIABLE      K50
3      VARIABLE      K0
4      VARIABLE      2*FN2
4      VARIABLE      FN2
4      VARIABLE      FN2
*      VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED      LAST ONE IS IN
5      VARIABLE      K250
5      VARIABLE      K1000      QUANTUM SIZE
5      VARIABLE      K1500      QUANTUM SIZE
5      VARIABLE      K2000      QUANTUM SIZE
5      VARIABLE      K500
*      VARIABLE      5 INDICATES THE SIZE      OF QUANTUM
*      IT IS CHANGED FOR DIFFERENT QUANTA,      THE LAST ONE IS IN EFFECT
6      VARIABLE      P10-V5      REDUCE REQUIRED TIME
7      VARIABLE      V7+1
8      VARIABLE      P9+1
9      VARIABLE      K5000
9      VARIABLE      K2500
9      VARIABLE      K2000
9      VARIABLE      K1666
9      VARIABLE      K1250
9      VARIABLE      K1000
9      VARIABLE      K800
9      VARIABLE      K1111
9      VARIABLE      K1111      MS
9      VARIABLE      K3333      MS
17     VARIABLE      K666      K3333/5
*      VARIABLE 9 IS 1/L INTER ARRIVAL TIME
*      VARIABLE 4 IS 1/U SERVICE TIME
*      V4/V9 = (1/U)/(1/L) = R.      UTILIZATION OF CPU
*      K1000 / K5000 = .2      TO VARY P FOR COND. WAITING
*      K1000 / K2500 = .4
*      K1000 / K1666 = .6
*      K1000 / K2000 = .5
*      K1000 / K1250 = .8
*      K1000 / K1111 = .9
*      K1000 / K1000 = 1.0
10     VARIABLE      P2+1
13     VARIABLE      (V4/V5)+1
GEN1   GENERATE      V9,K0,,,100
GEN1   GENERATE      V9,V9,,,100
GEN1   GENERATE      V9,FN1,,,100      DIFF PRIORITY
GEN1   GENERATE      V17,FN3,,,100
PAGE 2

```

```

ADVANCE 0,0
TRANSFER ,ADV89
GEN2 GENEPATE V9,KC,,,98
GEN2 GENFRATE V9,V9,,,98
GEN2 GENFRATE V9, FN1,,,98
GEN2 GENERATE V17, FN3,,,98
ADVANCE 0,0
TRANSFER ,ADV88
GEN3 GENERATE V9,KC,,,97
GEN3 GENERATE V9,V9,,,97
GEN3 GENERATE V9, FN1,,,97
GEN3 GENERATE V17, FN3,,,97
ADVANCE 0,0
ADVANCE 0,0
TRANSFER ,ADV88
ADV88 ADVANCE 0,0
ASSIGN 9,K1
ASSIGN 1C,V4 REQUIRED SERVICE TIME
ASSIGN 11,P10 GET THE REQ TIME FOR SAVE
QUE ADVANCE 0,0
FIF1 ASSIGN 2,V18
18 VARIABLE K1C1-PR
* IN DIFFERENT LEVELS ALL QUEUES IN TS RR HAVE SAME PRIORITY
QUEUE P2
LNK ADVANCE 0,0 TO BE
SOR ADVANCE 0,0
SEIZE CPU
DEPART P2
TES1 TRANSFER ,ADV3
ADV ADVANCE V5 QUANTUM
ADV0 ADVANCE 0,0
ADV3 ADVANCE P10 ADVANCE FOR THE R
ADV2 ADVANCE V3 OVERHEAD IN BATCH PROCESSING
RELEASE CPU
UNK ADVANCE 0,0 TO BE REPLACED
ASS1 ASSIGN 10,K0
TEST G P10,K0,END IS SERVICE COMPLETED ?
ASSIGN 9,V8
SPLIT 1,QUE
TERMINATE
END ADVANCE 0,0
TEST G P11,V5,END1
* WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
* MANY QUANTA
TABULATE XTM1
END1 ADVANCE 0,0
ASSIGN 6,M1 PUT TIME IN PARAMETER 6
14 VARIABLE (100*P11)/P6 C/C OF SERVICE OVER TIME IN SYSTEM
ASSIGN 7,V14
TABULATE XSER
TABULATE XTIME
TABULATE PASS
TERMINATE 1
XTIME TABLE M1,0,200,30
XTM1 TABLE M1,0,500,30
XSER TABLE P7,0,10,20 % OF SERVICE OVER TIME IN SYSTEM

```

PASS	TABLE	P9,0,1,60	DISTRIBUTION OF REQUIRED NUMBER	
	START	50,NP		TS120320
	RESET			TS120330
	START	100		
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	145,253	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		
	RESET			TS120330
	START	100		
	RMULT	631,759	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	135,527	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	585,889	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	439,037	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	033,283	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	449,301	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	683,277	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	829,651	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430

RL100440

```

      RESET
      START      100
      REPORT
      EJECT
      SPACE      3
FAC   TITLE      ,UTILIZATION OF THE CPU
      SPACE      3
QUE   TITLE      ,WAITING TIME FOR THE CPU
      SPACE      3
TAB   TITLE      ,TIME SPENT IN THE SYSTEM
      SPACE      3
      EJECT

```

```

*
  JOB      TS71
/*  END
/*  END
// JOB FIM01      2,2      TS70
  ALLOC F1=OK,F2=OK
// ASSGN SYSLOG,X'131'
*  MOUNT A WORK PACK ON 131
// DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD
// EXTENT SYSLOG,111111,1,0,0310,610
// DLBL INTER01,'SYSTEM WORK FILE NO. 0',69/032,SD
// EXTENT SYSLOG,111111,1,0,0310,610
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYSLOG,111112,8,1,2920,240,7
// DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYSLOG,111112,8,1,2920,240,7
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYSLOG,111112,8,1,2929,180,12
// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYSLOG,111112,8,1,2929,180,12
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYSLOG,111112,8,1,2933,180,19
// DLBL REPOG1,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYSLOG,111112,8,1,2933,180,19
// EXEC DARC1

```

SIMULATE

TS120010

***** NO 1

***** TIME SHARING

*** 1.1 NON-PREEMPTIVE ONLY ONE QUEUE

*** 1.2 NON-PREEMPTIVE ONE QUEUE PER PRIORITY

*** 1.3 NON-PREEMPTIVE LIMITED QUANTUM

*** A LIMITED QUANTUM IMPLIES A ROUND ROBIN MODEL IN EACH OF THE

*** QUEUE

*** THE PRIORITIES ARE NOT CHANGED AFTER EACH PASS

*** 2.1 PREEMPTIVE ONLY ONE QUEUE RECORDED

*** 2.2 PREEMPTIVE ONE QUEUE FOR EACH PRIORITY

*** 2.3 PREEMPTIVE WITH LIMITED QUANTUM

1 FUNCTION RN1,C24

0	0	.1	.104	.2	.222	.3	.355	.4	.509	.5	.69
.6	.915	.7	1.2	.75	1.38	.8	1.6	.84	1.83	.88	2.12
.9	2.3	.92	2.52	.94	2.81	.95	2.99	.96	3.2	.97	3.5

PAGE 5

```

.98 3.9 .99 4.6 .995 5.3 .998 6.2 .999 7 .9997 8
2 FUNCTION RN2,C24 EXP * 1000
0.000/.1,104/.2,222/.3,355/.4,509/.5,690/.6,915/.7,1220
.75,1380/.8,1600/.84,1830/.88,2120/.9,2300/.92,2520/.94,2810
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200
.999,7000/.9997,8000
* FN2 IS THE FN1 EXP TIMES 1000 FOR MORE PRECISION WHEN USING
* IN VARIABLE STATEMENT
* THE TIME IS EXPRESSED IN MILLISEC 1/1000 OF A SEC
3 FUNCTION RN1,C25 NORMAL DISTRIBUTION
0.000/0.00000,3.10/0.00135,2.0/0.00621,2.5/0.02275,3.0/0.06681,3.5
0.11507,3.7/0.15866,4.0/0.20186,4.2/0.27425,4.4/0.34458,4.6
0.42074,4.8/0.50000,5.0
0.57926,5.2/0.65542,5.4/0.72575,5.6/0.78814,5.8/0.84134,6.0
0.88493,6.2/0.93319,6.5/0.97725,7.0/0.99379,7.5/0.99865,8.0
0.99997,9.0/1.00000,10.0
1 VARIABLE 90 +30 AVERAGE OVERHEAD PER RECORD X NBR OF REC
2 VARIABLE 5 NBR OF RECORDS TO BE TRANSFERRED
2 VARIABLE 1
1 VARIABLE 100
1 VARIABLE 50
3 VARIABLE V1*V2 ACTUAL AVERAGE OVERHEAD PER QUANTUM
3 VARIABLE K50
3 VARIABLE K0
4 VARIABLE 2*FN2
4 VARIABLE FN2
4 VARIABLE FN2
* VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED LAST ONE IS IN
5 VARIABLE K250
5 VARIABLE K1000 QUANTUM SIZE
5 VARIABLE K1500 QUANTUM SIZE
5 VARIABLE K2000 QUANTUM SIZE
5 VARIABLE K500
* VARIABLE 5 INDICATES THE SIZE OF QUANTUM
* IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT
6 VARIABLE P10-V5 REDUCE REQUIRED TIME
7 VARIABLE V7+1
8 VARIABLE P9+1
9 VARIABLE K5000
9 VARIABLE K2500
9 VARIABLE K2000
9 VARIABLE K1666
9 VARIABLE K1250
9 VARIABLE K1000
9 VARIABLE K800
9 VARIABLE K1111
9 VARIABLE K1111 MS
9 VARIABLE K3333 MS
17 VARIABLE K666 K3333/5
* VARIABLE 9 IS 1/L INTER ARRIVAL TIME
* VARIABLE 4 IS 1/U SERVICE TIME
*  $V4/V9 = (1/U)/1/L = P$  UTILIZATION OF CPU
*  $K1000 / K5000 = .2$  TO VARY P FOR COND. WAITING
*  $K1000 / K2500 = .4$ 
*  $K1000 / K1666 = .6$ 
*  $K1000 / K2000 = .5$ 
PAGE 6

```



```

* K1000 / K1250 = .8
* K1000 / K1111 = .9
* K1000 / K1500 = 1.0
10 VARIABLE P2+1
13 VARIABLE (V4/V5)+1
GEN1 GENERATE V9,KC,,,100
GEN1 GENERATE V9,V9,,,100
GEN1 GENERATE V9,FN1,,,100 DIFF PRIORITY
GEN1 GENERATE V17,FN3,,,100
ADVANCE 0,0
TRANSFER ,ADV88
GEN2 GENERATE V9,KC,,,98
GEN2 GENERATE V9,V9,,,98
GEN2 GENERATE V9,FN1,,,98
GEN2 GENERATE V17,FN3,,,98
ADVANCE 0,0
TRANSFER ,ADV88
GEN3 GENERATE V9,KC,,,97
GEN3 GENERATE V9,V9,,,97
GEN3 GENERATE V9,FN1,,,97
GEN3 GENERATE V17,FN3,,,97
ADVANCE 0,0
ADVANCE 0,0
TRANSFER ,ADV88
ADV88 ADVANCE 0,0
ASSIGN 9,K1
ASSIGN 10,V4 REQUIRED SERVICE TIME
ASSIGN 11,P10 GET THE REQ TIME FOR SAVE
QUE ADVANCE 0,0
FIF1 ASSIGN 2,V18
18 VARIABLE K101-PR
* IN DIFFERENT LEVELS ALL QUEUES IN TS RR HAVE SAME PRIORITY
QUFUE P2
PREEMPT CPU,PR PREEMPT THE CPU EVEN IF IT IS IN USE
GATE U CPU,ADV99
TEST G PR,V1,RET
ADV99 ADVANCE 0,0
DEPART P2
TES0 TRANSFER ,ADV98
ADVANCE V5
ADV97 TRANSFER ,ADV96
ADV98 ADVANCE P10
ADV96 ADVANCE V3
RETURN CPU
TRANSFER ,UNK
RET RETURN CPU
TRANSFER ,LNK
LNK ADVANCE 0,0 TO BF
SOR ADVANCE 0,0
SEIZE CPU
SAVEVALUE 1,PR,XH SAVE THE PRIORITY OF THE CUT TX
DEPART P2
TES1 TRANSFER ,ADV3
ADV ADVANCE V5 QUANTUM
ADV0 ADVANCE 0,0
ADV3 ADVANCE P10 ADVANCE FOR THE R
PAGE 7

```

```

ADV2  ADVANCE  V3      OVERHEAD IN  BATCH PROCESSING
      SAVEVALUE 1,010,XH  FOR IF THERE IS NO TX      LOW PRIORITY
      RELEASE  CPU
UNK   ADVANCE  0,0  TO BE REPLACED
ASS1  ASSIGN   10,K0
      TEST G   P10,K0,END  IS SERVICE COMPLETED ?
      ASSIGN   9,V8
      SPLIT    1,QUE
      TERMINATE
END   ADVANCE  0,0
      TEST G   P11,V5,END1
*    WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
*    MANY QUANTA
      TABULATE XTM1
END1  ADVANCE  0,0
      ASSIGN   6,M1  PUT TIME IN PARAMETER 6
      14  VARIABLE (100*P11)/P6  0/0 OF SERVICE OVER TIME IN SYSTEM
      ASSIGN   7,V14
      TABULATE XSER
      TABULATE XTIME
      TABULATE PASS
      TERMINATE 1
XTIME TABLE M1,0,200,30
XTM1  TABLE M1,0,500,30
XSER  TABLE P7,0,10,20  % OF SERVICE OVER TIME IN SYSTEM
PASS  TABLE P9,0,1,60  DISTRIBUTION OF REQUIRED NUMBER
      START    50,NP
      RESET
      START    100
      CLEAR
      START    50,NP
      RESET
      START    100
      RMULT    145,253  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START    50,NP
      RESET
      START    100
      RMULT    631,759  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START    50,NP
      RESET
      START    100
      RMULT    135,527  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START    50,NP
      RESET
      START    100
      RMULT    585,899  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START    50,NP
      RESET
      START    100
      RMULT    439,637  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START    50,NP

```

TS120320
TS120330

RL100430
RL100440

TS120330

RL100430
RL100440

RL100430
RL100440

RL100430
RL100440

	RESET			RL100440
	START	100		
	RMULT	033,283	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	449,301	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	683,277	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	RMULT	829,651	CHANGE THE SEED FOR NEW RUNS	
	CLEAR			
	START	50,NP		RL100430
	RESET			RL100440
	START	100		
	REPORT			
	EJECT			
	SPACE	3		
FAC	TITLE		,UTILIZATION OF THE CPU	
	SPACE	3		
QUE	TITLE		,WAITING TIME FOR THE CPU	
	SPACE	3		
TAB	TITLE		,TIME SPENT IN THE SYSTEM	
	SPACE	3		
	EJECT			
*				
	JOB	TS71		
/*	END			
/C	END			
	END OF DATA			
PAGE	9			

*** APPENDIX NINE ***

PROGRAM TS70 TEST OF TIME SHARING
AUTHOR: PROSPER M. BERNARD

```
// JOB TS70 TIME SHARING MODEL
// ALLOC F1=OK,F2=OK
// ASSIGN SYS000,X'131'
* MOUNT A WORK PACK ON 131
// DLBL IJSYS00,'SYSTEM WORK FILE NO. 0',68/032,SD
// EXTENT SYS000,111111,1,0,0310,610
// DLBL INTER01,'SYSTEM WORK FILE NO. 0',68/032,SD
// EXTENT SYS000,111111,1,0,0310,610
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYS001,111112,8,1,2920,240,7
// DLBL SIMIN11,'SYSTEM WORK FILE NO. 1',67/365,SD
// EXTENT SYS001,111112,8,1,2920,240,7
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYS002,111112,8,1,2929,180,12
// DLBL SYMTAB1,'SYSTEM WORK FILE NO. 2',67/365,SD
// EXTENT SYS002,111112,8,1,2929,180,12
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYS003,111112,8,1,2933,180,19
// DLBL REPORG1,'SYSTEM WORK FILE NO. 3',67/365,SD
// EXTENT SYS003,111112,8,1,2933,180,19
// EXEC DARG1
```

REALLOCATE QUE,110,XAC,100,FSV,7,MAC,0,LOG,20
SIMULATE

TS120010

***** NO 1

**** THE ADIRI AND AVI-ITZHAK MODEL
**** FOR A FIXED NUMBER OF USERS

**** 1 VARYING QUANTUM
**** 2 VARYING NUMBER OF USERS
**** 3 VARYING SERVICE REQUIREMENT

***** ALL TIMES ARE EXPRESSED IN MILI SECONDS

1 FUNCTION RN1,C24											
0	0	.1	.104	.2	.222	.3	.355	.4	.509	.5	.69
.6	.915	.7	1.2	.75	1.38	.8	1.6	.84	1.83	.88	2.12
.9	2.3	.92	2.52	.94	2.81	.95	2.99	.96	3.2	.97	3.5
.98	3.9	.99	4.6	.995	5.3	.998	6.2	.999	7	.9997	8
2 FUNCTION RN2,C24 EXP * 1000											
0,000/.1,104/.2,222/.3,355/.4,509/.5,697/.6,915/.7,1220											
.75,1380/.8,1620/.84,1837/.88,2120/.9,2300/.92,2520/.94,2810											
.95,2999/.96,3200/.97,3500/.98,3950/.99,4600/.995,5300/.998,6200											
.999,7000/.9997,8000											

* FN2 IS THE FN1 EXP TIMES 1000 FOR MORE PRECISION WHEN USING
* IN VARIABLE STATEMENT

3 FUNCTION RN1,C25 NORMAL DISTRIBUTION

PAGE 1

0.0/0.00003,1.0/0.00135,2.0/0.00621,2.5/0.02275,3.0/0.06681,3.5
 0.11507,3.7/0.15866,4.0/0.21186,4.2/0.27425,4.4/0.34458,4.6
 0.42074,4.8/0.50000,5.0
 0.57926,5.2/0.65542,5.4/0.72575,5.6/0.78814,5.8/0.84134,6.0
 0.88493,6.2/0.93319,6.5/0.97725,7.0/0.99379,7.5/0.99865,8.0
 0.99997,9.0/1.00000,10.0

3	VARIABLE	K2	.2	SEC	OVERHEAD
3	VARIABLE	K10	1.0	SEC	OVERHEAD
3	VARIABLE	K0			
3	VARIABLE	K1	01	SEC	OVERHEAD
4	FVARIABLE	FN1*1800	180	SEC	
4	FVARIABLE	FN1*1200	120	SEC	
4	FVARIABLE	FN1*10		1 SEC	
4	FVARIABLE	FN1*600	60	SEC	S
4	FVARIABLE	FN1*1000	100		U=

* VARIABLE NO 4 IS FOR TIME OF SERVICE NEEDED LAST ONE IS IN

5	VARIABLE	K2	0.2	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K3	0.3	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K4	0.4	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K5	0.5	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K6	0.6	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K7	0.7	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K8	0.8	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K9	0.9	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K11	1.1	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K12	1.2	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K20	2.0	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K200	20.0		QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K1	0.1	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K13	1.3	SEC	QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K100	QUANTUM SIZE	EXCL	OVERHEAD
5	VARIABLE	K100	10.0		QUANTUM SIZE EXCL OVERH.
5	VARIABLE	K10	1.0	SEC	QUANTUM SIZE EXCL OVERH.
6	VARIABLE	P10-V5	REDUCE	REQUIRED TIME	
7	VARIABLE	V7+1			

* IT IS CHANGED FOR DIFFERENT QUANTA, THE LAST ONE IS IN EFFECT

* VARIABLE 5 INDICATES THE SIZE OF QUANTUM

8	VARIABLE	P9+1			
9	VARIABLE	K10000	NL/U = 1.0, N=10	U=0.01/SEC	
9	VARIABLE	K11764	NU/U = 0.85 N=10	U=0.01/SEC	
9	VARIABLE	K20000	NL/U = 0.5, N=10	U=0.01/SEC	
9	VARIABLE	K5000	NL/U = 1.0, N= 5	U=0.01/SEC	
9	VARIABLE	K10000	NL/U = 0.5, N= 5	U=0.01/SEC	
17	VARIABLE	K4000	K20000/5		
17	VARIABLE	K2000	K10000/5		
17	VARIABLE	K1000	K5000/5		
17	VARIABLE	K2353	K11764/5		

 **** NOTE: NL/U= 0.5 THIS IS UTILIZATION ONLY IF OVERHEAD= 0 ****

* VARIABLE 4 IS 1/U SERVICE TIME
 * VARIABLE 9 IS 1/(NL) L = ARRIVAL RATE, N= NUMBER OF USERS
 * $V4/V9 = (1/U)/(1/L) = R$. UTILIZATION OF CPU
 * $K1000 / K5000 = .2$ TO VARY P FOR COND. WAITING
 * $K1000 / K2500 = .4$
 * $K1000 / K1660 = .6$

```

* K1000 / K2000 = .5
* K1000 / K1250 = .8
* K1000 / K1111 = .9
* K1000 / K1000 = 1.0
10 VARIABLE P2+1
10 VARIABLE K101-PR          TO DETERMINE WHICH ONE IT SHOULD BE
16 VARIABLE PR              NOT TO CHANGE PRIORITY FOR RR
16 VARIABLE PR-1
13 VARIABLE (V4/V5)+1
14 VARIABLE (1+P11)/P6      O/O OF SERVICE OVER TIME IN SYSTEM
15 VARIABLE K100            NUMBER OF USERS
15 VARIABLE K50             NUMBER OF USERS
15 VARIABLE K20             NUMBER OF USERS
15 VARIABLE K5              NUMBER OF USERS
15 VARIABLE K1              NUMBER OF USERS
15 VARIABLE K3              NUMBER OF USE
15 VARIABLE K10             NUMBER OF U
18 VARIABLE QT1+QT2+QT3
19 VARIABLE V19+QT4+QT5+QT6+QT7
20 VARIABLE K50             LIMITS ON THE NUMBER OF LEVELS
20 VARIABLE K100            LIMITS ON THE NUMBER OF LEVELS
20 VARIABLE K25             LIMITS
20 VARIABLE K25             LIMITS ON THE NUMBER OF LEVES
20 VARIABLE K250
20 VARIABLE K10             LIMITS ON THE NUMBER OF LEVELS
20 VARIABLE K1              LIMITS ON THE NUMBER OF LEVEL

```

```

*****
***                                     ***
***                                     ***
***                                     ***
***                                     ***
*** ALL TIMES ARE EXPRESSED IN 1/10 OF A SECOND ***
*****

```

```

GEN1 GENERATE V9,0,,,100,,F      CONSTANT
GEN1 GENERATE V9,V9,,,100,,F      UNIFORM
GEN1 GENERATE V17,FN1,,,100,,F    NORMAL
GEN1 GENERATE V9,FN1,,,100,,F
      ADVANCE 0,0
      TRANSFER ,ADV88
GEN2 GENERATE V9,0,,,100,,F      CONSTANT
GEN2 GENERATE V17,FN1,,,100,,F    NOR
GEN2 GENERATE V9,V9,,,100,,F      UNIFORM
GEN2 GENERATE V9,FN1,,,100,,F
      ADVANCE 0,0
      TRANSFER ,ADV88
GEN3 GENERATE V9,0,,,100,,F      CONSTANT
GEN3 GENERATE V17,FN1,,,100,,F    NOR
GEN3 GENERATE V9,V9,,,100,,F      UNIFORM
GEN3 GENERATE V9,FN1,,,100,,F
      ADVANCE 0,0
      TRANSFER ,ADV88
GEN4 GENERATE V17,FN1,,,100,,F    NOR
GEN4 GENERATE V9,V9,,,100,,F      UNIFORM
GEN4 GENERATE V9,0,,,100,,F      CONSTANT
GEN4 GENERATE V9,FN1,,,100,,F
      ADVANCE 0,0

```

	TRANSFER	,ADV88	
GEN5	GENERATE	V17,FN1,,,100,,F	N
GEN5	GENERATE	V9,C,,,100,,F	CONSTANT
GEN5	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN5	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	
	TRANSFER	,ADV88	
GEN6	GENERATE	V17,FN1,,,100,,F	N
GEN6	GENERATE	V9,C,,,100,,F	CONSTANT
GEN6	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN6	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	<
	TRANSFER	,ADV88	
GEN7	GENERATE	V17,FN1,,,100,,F	N
GEN7	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN7	GENERATE	V9,C,,,100,,F	CONSTANT
GEN7	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	
	TRANSFER	,ADV88	
GEN8	GENERATE	V17,FN1,,,100,,F	N
GEN8	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN8	GENERATE	V9,C,,,100,,F	CONSTANT
GEN8	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	
	TRANSFER	,ADV88	
GEN9	GENERATE	V17,FN1,,,100,,F	N
GEN9	GENERATE	V9,C,,,100,,F	CONSTANT
GEN9	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN9	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	
	TRANSFER	,ADV88	
GEN10	GENERATE	V9,V9,,,100,,F	UNIFORM
GEN10	GENERATE	V9,C,,,100,,F	CONSTANT
GEN10	GENERATE	V17,FN1,,,100,,F	NORMAL
GEN10	GENERATE	V9,FN1,,,100,,F	
	ADVANCE	0,0	
	TRANSFER	,ADV88	
ADV88	ADVANCE	0,0	
	ASSIGN	9,K1	
	ASSIGN	10,V4	REQUIRED SERVICE TIME
	ASSIGN	11,P10	GET THE REQ TIME FOR SAVE
QUE	ADVANCE	0,0	
FIF1	ASSIGN	2,V10	FOR NON FIFO
*	IN DIFFERENT	LEVELS	ALL QUEUES IN TS RR HAVE SAME PRIORITY
	QUEUE	P2	
LNK	ADVANCE	0,0	TO BE
SOR	ADVANCE	0,0	
	SEIZE	CPU	
	DEPART	P2	
TFS1	TEST G	P10,V5,ADV3	
ADV	ADVANCE	V5	QUANTUM
ADV0	TRANSFER	,ADV2	
ADV3	ADVANCE	P10	ADVANCE FOR THE R
ADV2	ADVANCE	V3	OVERHEAD IN BATCH PROCESSING
	RELEASE	CPU	
UNK	ADVANCE	0,0	TO BE REPLACED

```

ASS1  ASSIGN      10,V6
      TEST G      P10,K7,END  IS SERVICE COMPLETED ?
      ASSIGN      9,V8
      TEST L      P2,V20,FIF2          TO TEST THE LIMIT ON NBR OF LEVE
      PRIORITY    V16
FIF2  ADVANCE     0,0
      SPLIT       1,QUE
      TERMINATE
END   ADVANCE     0,0
      TABULATE    XTIME
      TABULATE    PASS
      SAVEVALUE   1,QT1
      SAVEVALUE   3,V18
      SAVEVALUE   7,V19
      TEST L      P11,K400,END40.  TO TEST FOR RESPONSE TIME
      TABULATE    XTIM2
      TRANSFER    ,ADV66
END40 ADVANCE     0,0
      TEST L      P11,K800,END80
      TABULATE    XTIM3
      TRANSFER    ,ADV66
END80 ADVANCE     0,0
      TEST L      P11,K1200,END12
      TABULATE    XTIM4
      TRANSFER    ,ADV66
END12 ADVANCE     0,0
      TEST L      P11,K1600,END16
      TABULATE    XTIM5
      TRANSFER    ,ADV66
END16 ADVANCE     0,0
      TABULATE    XTIM6
ADV66 ADVANCE     0,0
      TEST G      P11,V5,END1
*   WE TABULATE THE TRANSIT TIME FOR TRANSACTION REQUIRING
*   MANY QUANTA
      TABULATE    XTIM1
END1  ADVANCE     0,0
      ASSIGN      6,M1  PUT TIME IN PARAMETER 6
      ASSIGN      7,V14
      TABULATE    XSER
      TERMINATE   1
XTIME TABLE     M1,0,1000,50
XTIM1 TABLE     M1,0,1000,50
XTIM2 TABLE     M1,0,1000,50
XTIM3 TABLE     M1,0,1000,50
XTIM4 TABLE     M1,0,1000,50
XTIM5 TABLE     M1,0,1000,50
XTIM6 TABLE     M1,0,1000,50
XSER  TABLE     P7,0,10,20  % OF SERVICE OVER TIME IN SYSTEM
PASS  TABLE     P9,0,1,60  DISTRIBUTION OF REQUIRED NUMBER
      START
      RESET
      START      100
      RMULT      145,253  CHANGE THE SEED FOR NEW RUNS
      CLEAR
      START      50,NP

```

TS120327
TS120330

RL100430

RESET			RL100440
START	100		
RMULT	821,805	CHANGE THE SEED FOR-	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	631,759	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	135,527	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	585,889	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	439,037	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		
RESET			
START	100		
RMULT	033,283	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	449,301	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	683,277	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	623,419	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
RMULT	303,189	CHANGE THE SEED FOR NEW RUNS	
CLEAR			
START	50,NP		RL100430
RESET			RL100440
START	100		
REPORT			
EJECT			
SPACE	3		
TITLE	UTILIZATION OF THE CPU		

	SPACE	3
QUE	TITLE	,WAITING TIME FOR THE CPU
	SPACE	3
TAB	TITLE	,TIME SPENT IN THE SYSTEM
	SPACE	3
	SPACE	3
SAV	TITLE	,TOTAL WAITING TIME IN QUEUE
	EJECT	

*
END
/* END OF DATA
/& END
END OF DATA
PAGE 7

Proof of the RR waiting time¹ (eq. 6.1.1)

We consider a unit (which we call the "tagged" unit) arriving at the RR system in equilibrium and assume a service requirement of t sec. Defining k as the smallest integer such that $t < kq$, we address the problem of finding the tagged unit's average waiting in queue. To find the mean wait in system we simply add t to the waiting time in queue.

Assume that on arrival of the tagged unit there is one or no unit in service and n in the queue. We decompose the waiting time in queue into two parts, T_1 and T_2 . T_1 corresponds to the time required to finish the unit, if any, in service (taking into account the possibility of its returning for more) plus the time required to process (not necessarily to completion) all arrivals during this time. T_2 corresponds to the time required to properly service the n units in the queue at arrival. Of course, both T_1 and T_2 must take into consideration the processing of all arrivals that occur in T_1 and T_2 . Evidently, the mean waiting time in queue is

$$W_k = E(T_1) + E(T_2). \quad (\text{A.1})$$

The resequencing of events implicit in our definitions will clearly not affect the determination of W_k so long as all events are taken into account. This often-used "resequencing" approach is justified by the fact that the input process is time-homogeneous and statistically independent of the state of the system.

Now for $E(T_2)$ we use expected value arguments essentially the same as those used by Kleinrock² for the discrete system. Let y_i denote the time spent in queue on the i th pass by the tagged unit. Since the tagged unit must make k passes we may write

$$E(T_2) = E\left\{\sum_{i=1}^k y_i\right\} = \sum_{i=1}^k E(y_i). \quad (\text{A.2})$$

Correspondingly, we define N_i as the mean number of units ahead of the tagged unit at the beginning of the i th pass. We now develop a general expression for N_i . For $i > 1$, N_i will be composed of the mean number of those units of N_{i-1} whose service requirements exceed q sec (we call these returning units), and the mean number of new arrivals that occur during the time interval $y_{i-1} + q$. (The q sec is included because of the tagged unit's service following y_{i-1} .)

From the memory-less property of the exponential distribution we may observe that the probability δ with which a unit returns (requires more than q sec of service) is independent of i and given by

$$\delta = \int_q^\infty \mu e^{-\mu\tau} d\tau = e^{-\mu q}. \quad (\text{A.3})$$

Thus, we have

$$N_i = \delta N_{i-1} + \lambda[E(y_{i-1}) + q]. \quad (\text{A.4})$$

But

$$E(y_{i-1}) = N_{i-1}E_1(\tau),$$

¹ The proof is reproduced here with the written permission of the Association for Computing Machinery. It is taken from: Edward Coffman and Leonard Kleinrock. "Feedback Queuing Models for Time-Shared Systems." Journal of ACM, Vol. 15, No. 4, October 1968, p. 567.

² Kleinrock, L. Analysis of a Time-Shared Processor. Nav. Res. Logistics Quart. 11, 10 (March 1964), pp. 59-73.

so upon substitution in eq. (A.4) we obtain

$$N_i = N_{i-1}[\delta + \lambda E_1(\tau)] + \lambda q. \quad (\text{A.5})$$

For convenience we define

$$\beta = \delta + \lambda E_1(\tau), \quad (\text{A.6})$$

so that

$$N_i = \beta N_{i-1} + \lambda q. \quad (\text{A.7})$$

Now solving this equation for N with the condition $N_1 = \bar{n} = E(n)$ yields

$$N_i = \beta^{i-1} \bar{n} + \lambda q \sum_{j=0}^{i-2} \beta^j, \quad i > 1. \quad (\text{A.8})$$

Using induction eq. (A.8) is easily established. From eq. (A.2) we may now write

$$E(T_2) = E_1(\tau) \sum_{i=1}^k N_i, \quad (\text{A.9})$$

whereupon substitution of eq. (A.8) into eq. (A.9) yields, after carrying out the summations

$$E(T_2) = \frac{E_1(\tau)}{1-\beta} \left[\lambda k q + \left(\bar{n} - \frac{\lambda q}{1-\beta} \right) (1-\beta^k) \right], \quad (\text{A.10})$$

where, by evaluating eq. (A.6), we have

$$\beta = \rho + (1-\rho)e^{-\mu q}.$$

Now in the RR and FB_N models we have assumed that no losses or "overhead" times exist in system operation, and in both models no advantage is taken of any a priori information concerning the nature of the new arrivals. Thus, it is not difficult to see that the average number of units in the queue for both the RR and FB_N systems is precisely the same as for the exponential FCFS (Erlang's) system. Thus, we may solve for $\bar{n}$ by using the corresponding result for the FCFS system which is given by [12]³

$$\bar{n} = \frac{\rho^2}{1-\rho}$$

(in queue). Now using $E_1(\tau) = (1/\mu)[1 - e^{-\mu q}]$, we may render eq. (A.10) as

$$E(T_2) = \frac{(1/\mu)}{1-\rho} \left[\lambda k q + \left(\frac{\rho^2}{1-\rho} - \frac{\lambda q}{1-\beta} \right) (1-\beta^k) \right]. \quad (\text{A.11})$$

Turning now to $E(T_1)$, let W_0 be the mean amount of time required to complete the quantum-service in progress at the time of arrival. Then $E(T_1)$ is equal to W_0 plus the expected time to process the mean number of arrivals in W_0 plus the time it takes to process the unit in service if it returns for more service. Here again, the processing referred to includes the processing of subsequent arrivals as for $E(T_2)$. The mean number of arrivals in W_0 is given by λW_0 . If we call σ the probability that the unit in service at arrival returns for more service we have

$$\bar{n}' = \sigma + \lambda W_0 \quad (\text{A.12})$$

³ Equation 12 is the same as equation 6.1.01

as the mean number of units (excluding $\bar{n}$) to service following W_0 . The time to process the $\bar{n}'$ units can be calculated as for $E(T_2)$. We note, however, that these units are all "behind" the tagged unit and therefore will be provided with a maximum of only $(k - 1)$ quanta of service before the tagged unit receives its last quantum. Thus, we can proceed as before and form the sum

$$N_1 E_1(\tau) + [\delta N_1 + \lambda N_1 E_1(\tau)] E_1(\tau) + \cdots + [\delta N_{k-2} + \lambda N_{k-2} E_1(\tau)] E_1(\tau),$$

from which it is easy to establish by induction and by using $N_1 = \bar{n}'$,

$$\sum_{i=1}^{k-1} N_i = \frac{1 - \beta^{k-1}}{1 - \beta} \bar{n}'. \quad (\text{A.13})$$

Finally, therefore, we have

$$E(T_1) = W_0 + [\sigma + \lambda W_0] \left[\frac{1 - \beta^{k-1}}{1 - \beta} \right] E_1(\tau). \quad (\text{A.14})$$

Using $E_1(\tau) = (1/\mu)[1 - e^{-\mu\tau}]$ from eq. (12), this may be put into the form

$$E(T_1) = \frac{W_0}{1 - \rho} [1 - \rho\beta^{k-1}] + \frac{\sigma(1/\mu)}{1 - \rho} [1 - \beta^{k-1}]. \quad (\text{A.15})$$

It remains to derive expressions for W_0 and σ . To find W_0 we follow Cobham [1] and observe the following. Given that a quantum-service of duration t is in progress at the time of the tagged unit's arrival, then from the point of view of the unit being served the expected time of arrival is simply $(t/2)$. We must now determine the probability $dC(t)$ of arriving when a quantum-service of duration t is in progress. For this Cobham writes

$$dC(t) = \lambda_q t dF_1(t), \quad (\text{A.16})$$

where $F_1(t)$ is the quantum-service distribution given by eq. (12) and λ_q represents the average arrival rate of quantum services. Now eq. (A.16) is based on a Poisson arrival mechanism of quantum-services; in our case *unit* arrivals are Poisson which gives rise to Poisson "bulk" arrivals of quantum-services. However, eq. (A.16) still applies since for our purposes only the randomness or Poisson nature of the arrival times is necessary for eq. (A.16). Since a unit requires a k th pass (quantum-service) with probability $e^{-\mu(k-1)q}$ we see that

$$\lambda_q = \lambda \sum_{k=1}^{\infty} e^{-\mu(k-1)q} = \frac{\lambda}{1 - e^{-\mu q}}. \quad (\text{A.17})$$

Therefore, we obtain with Cobham

$$W_0 = \int_0^{\infty} \frac{t}{2} dC(t) = \frac{\lambda/2}{1 - e^{-\mu q}} E_1(t^2). \quad (\text{A.18})$$

To determine σ we find the probability that the tagged unit arrives and finds a program which will return for more service following the current quantum. Equivalently, we want the probability of an arrival during a quantum-service of exactly q sec in length. From eq. (A.16) and eq. (12) for $dF_1(t)$ we find

$$\sigma = \lambda_q q dF_1(q) = \lambda_q q e^{-\mu q}. \quad (\text{A.19})$$

Now from eq. (A.17) we get the following result:

$$\sigma = \frac{\lambda q \epsilon^{-\mu q}}{1 - \epsilon^{-\mu q}}. \quad (\text{A.20})$$

Inserting eqs. (A.18) and (A.20) into eq. (A.15), we get

$$E(T_1) = \frac{[(\lambda/2)/(1 - \epsilon^{-\mu q})]E_1(\tau^2)}{1 - \rho} [1 - \rho\beta^{k-1}] + \frac{\lambda q \epsilon^{-\mu q}}{1 - \beta} [1 - \beta^{k-1}] \cdot \frac{1}{\mu}, \quad (\text{A.21})$$

where

$$\beta = \rho + (1 - \rho)\epsilon^{-\mu q}. \quad (\text{A.22})$$

Substituting eqs. (A.21) and (A.11) into eq. (A.1) now yields

$$\begin{aligned} W(t) = & \frac{\rho k q}{1 - \rho} + \frac{(\lambda/2)E_1(\tau^2)}{1 - \beta} [1 - \rho\beta^{k-1}] \\ & + \frac{1}{1 - \rho} \left[\frac{\rho^2}{1 - \rho} \frac{1}{\mu} - \frac{\rho q}{1 - \beta} \right] [1 - \beta^k] + \frac{\lambda q \epsilon^{-\mu q}}{1 - \beta} \frac{1}{\mu} [1 - \beta^{k-1}], \end{aligned} \quad (\text{A.23})$$

which constitutes the result of Theorem when the service time t is added. Q.E.D.

We may now produce the result for the processor-shared model of Theorem 3 by taking the limit of eq. (A.23) as q goes to zero. Since the waiting time is conditioned on the service required, we want to hold kq constant while allowing q to go to zero in eq. (A.23). Calling $kq = t$, let us first calculate

$$\lim_{q \rightarrow 0} \beta^k = \lim_{q \rightarrow 0} [\rho + (1 - \rho)\delta]^k, \quad \delta = \epsilon^{-\mu q}.$$

With rearrangement we have

$$\beta^k = \sum_{i=0}^k \binom{k}{i} \delta^i [\rho(1 - \delta)]^{k-i} = \delta^k + k\rho(1 - \delta)\delta^{k-1} + \frac{k(k-1)}{2} \rho^2(1 - \delta)^2\delta^{k-2} \dots$$

Now $kq = t$ implies $\delta^k = \epsilon^{-\mu t}$, and approximating $(1 - \epsilon^{-\mu q})$ by μq for $0 < q \ll 1$ we have

$$\lim_{q \rightarrow 0} \beta^k = \epsilon^{-\mu t} \left[1 + \rho\mu t + \frac{(\rho\mu t)^2}{2!} + \dots \right] = \epsilon^{-\mu t(1-\rho)}.$$

With the same approximation it is easy to establish

$$\lim_{q \rightarrow 0} \frac{\lambda q}{1 - \delta} = \rho, \quad \lim_{q \rightarrow 0} \frac{E_1(\tau^2)}{1 - \delta} = 0,$$

so that on substitution of the above limits into eq. (A.23) we get

$$\lim_{q \rightarrow 0} W(t) = \frac{1}{1 - \rho} \left\{ \rho t - \frac{\rho}{\mu} [1 - \epsilon^{-\mu t(1-\rho)}] + \frac{\rho}{\mu} [1 - \epsilon^{-\mu t(1-\rho)}] \right\} = \frac{\rho t}{1 - \rho},$$

which establishes Theorem 3 after adding the service requirement (t) .

⁴ Theorem 3 is the same as equation 6.1.2

APPENDIX ELEVEN

Proof of the Waiting-Time theorem

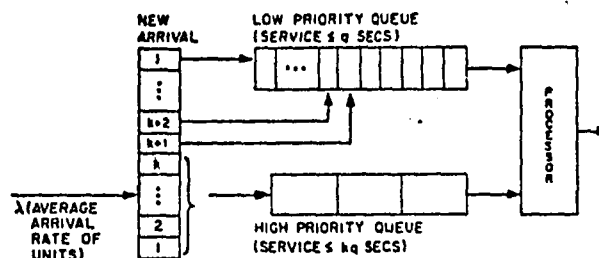
Feedback Model
(ed. 6.1.5)

FIG. 12. Equivalent two-level model without feedback

We consider a unit (the tagged unit) arriving at the FB_N system in equilibrium, assume that its service requirement is t sec, and define k as the smallest integer such that $kq > t$. We break up the waiting time in queue into two parts so that we may write

$$W_k = E(T_1) + E(T_2), \quad (B.1)$$

where T_1 is the time to complete the unit in service plus the time required to process the units which were in the first k queues at the time of arrival, and T_2 is the time to process all new arrivals that occur during the tagged unit's waiting time.

We approach the problem of determining W_k for $k < N$ by looking at a special two-level model which is equivalent in the sense of the waiting time we seek. Figure 12 shows this equivalent two-level model. Note from the figure that arrivals requiring j quanta of service are (artificially) separated into j corresponding parts. The first k parts (or j parts if $k > j$) are combined into a single arrival unit to the high-priority (lower level) queue. The remaining parts, if any, each constitute a unit arrival to the low-priority queue. In this special model "feedback" is no longer explicit. Indeed, the quantum-at-a-time processing is no longer carried out by the server, but is implicit in the arrival processing mechanism instead. However, for the waiting times of high-priority arrivals (requiring kq sec or less) for which feedback does not exist anyway, it is clear that this artificial arrival mechanism has not changed anything. As can be observed, arrivals to the high-priority queue are Poisson while arrivals to the low-priority queue are Poisson in "bulk."

From the above remarks we now make the simplifying observation that the time (T_1) to process the first k queues in the FB_N model ($N > k$) and the unit in service at arrival is the same as the waiting time in the high-priority queue of the special two-level model in Figure 12. In both cases the tagged unit must wait through the processing of units being allocated kq sec of service. It remains, therefore, to determine the high-priority waiting time of the special two-level model. But for this statistic we may identify our special two-level model with the corresponding single-channel, head-of-the-line (two-level) priority model of Cobham². The only difference we make between these two two-level models is that in the latter the arrival process to the low-priority queue is assumed to be Poisson instead of Poisson in bulk. But for the average waiting time in the *high-priority* queue it is unimportant whether or not the arrival process to the low-priority queue is Poisson. Indeed, it can be shown that the high-priority waiting-time distribution depends on the low-priority arrival process only through its average rate. Thus, using Cobham's result for the high-priority average waiting time we have

1 Reproduced with written permission from the Association for Computing Machinery. It is taken from: Edward Coffman and Leonard Kleinrock, "Feedback Queuing Models for Time-Shared Systems," *Journal of ACM*, Vol. 15, No. 4, October 1968, p. 571.

2 Cobham A. Priority assignment in waiting-line problems. *Oper. Res.*

$$E(T_1) = \frac{W_0}{1 - \rho_1}, \quad (\text{B.2})$$

where ρ_1 is the utilization factor for the high-priority queue and W_0 is the average amount of time required to finish the unit being served at the time of arrival. In our case

$$1 - \rho_1 = 1 - \lambda E_k(\tau) = 1 - \rho(1 - \epsilon^{-\mu k q}), \quad (\text{B.3})$$

where $E_k(\tau)$ is given by eq. (24), and

$$W_0 = \frac{\lambda_k}{2} \int_0^\infty \tau^2 dF_k(\tau) + \frac{\lambda_1}{2} \int_0^\infty \tau^2 dF_1(\tau), \quad (\text{B.4})$$

where λ_k , λ_1 and $F_k(\tau)$, $F_1(\tau)$ represent, respectively, the average arrival rates and service time distributions for the high-priority and low-priority queues. The distributions are defined by eq. 6.1.5. Now since an arrival requires service at the low-priority queue only if it requires in excess of kq sec of service we have

$$\lambda_k = \lambda, \quad \lambda_1 = \lambda \sum_{j=k}^{\infty} \epsilon^{-\mu j q} = \frac{\lambda \epsilon^{-\mu k q}}{1 - \epsilon^{-\mu q}} = \gamma_k \lambda. \quad (\text{B.5})$$

Thus,

$$E(T_1) = \frac{(\lambda/2)[E_k(\tau^2) + \gamma_k E_1(\tau^2)]}{1 - \rho(1 - \epsilon^{-\mu k q})}. \quad (\text{B.6})$$

To calculate $E(T_2)$ we now return to the original FB_N model. We observe that the average number of arrivals in W_k must be based on $W_k + (k - 1)q$ since the tagged unit received $(k - 1)q$ sec of service before reaching the k th (less than N th) queue. Clearly, each of the new arrivals must be allocated $(k - 1)$ quanta of service of which $E_{k-1}(\tau)$ is the average amount taken. Thus,

$$E(T_2) = \lambda[W_k + (k - 1)q]E_{k-1}(\tau). \quad (\text{B.7})$$

Finally, therefore,

$$W_k = \lambda[W_k + (k - 1)q]E_{k-1}(\tau) + \frac{(\lambda/2)[E_k(\tau^2) + \gamma_k E_1(\tau^2)]}{1 - \rho(1 - \epsilon^{-\mu k q})}, \quad 1 \leq k \leq N - 1. \quad (\text{B.8})$$

Solving for W_k and substituting for $E_{k-1}(\tau)$, we obtain

$$W_k = \frac{(\lambda/2)[E_k(\tau^2) + \gamma_k E_1(\tau^2)]}{[1 - \rho(1 - \epsilon^{-\mu k q})][1 - \rho(1 - \epsilon^{-\mu(k-1)q})]} + \frac{\rho(1 - \epsilon^{-\mu(k-1)q})}{1 - \rho(1 - \epsilon^{-\mu(k-1)q})} (k - 1)q, \quad 1 \leq k \leq N - 1. \quad (\text{B.9})$$

Adding t to eq. (B.9) now produces eq. 6.1.5.

Finally, for $k > N - 1$ we may simplify matters by observing that all units in the system at the time of arrival must be served to completion before the tagged unit comes to the service point for the N th time. Thus for $E(T_1)$ we may use the result for the waiting time in queue for the FCFS system. In particular, from eq. 6.1.4,

we have (subtracting the time t in the server)

$$E(T_1) = \frac{\rho(1/\mu)}{1 - \rho}. \quad (\text{B.10})$$

Now the period during which we must allow for new arrivals is again $W_k + (k - 1)q$. Because of the nature of the N th queue each of these new arrivals will be allocated $(N - 1)q$ sec of service. Thus

$$E(T_2) = \lambda[W_k + (k - 1)q]E_{N-1}(\tau). \quad (\text{B.11})$$

Adding eqs. (B.10) and (B.11) and solving for W_k now yields

$$W_k = \frac{\rho(1/\mu)}{(1 - \rho)[1 - \rho(1 - e^{-\mu(N-1)q})]} + \frac{\rho(1 - e^{-\mu(N-1)q})}{1 - \rho(1 - e^{-\mu(N-1)q})} (k - 1)q, \quad k \geq N. \quad (\text{B.12})$$

Adding t to eq. (B.12) now establishes eq. (22b) of Theorem 5 and completes the proof of Theorem 5. Q.E.D.

APPENDIX TWELVE

Proof of the SJF Waiting Time Theorem.¹
(eq. 6.1.10)

Cobham has has studied some priority problems in waiting lines.² Cobham's results are used here as a basis. However the priorities are continuous since they are assigned by the length of the service requirement.

The 'company' is assumed to own an infinite number of machines, subject to random (Poisson law) breakdown. A steady state is considered in which the number of operable machines is infinite. In this steady state new repair jobs of various magnitudes are generated at an average arrival rate λ . No specialized assumption is made about the distribution of repair times. Repair of random failures alone is considered; necessary checks and routine preventive maintenance of a periodic, predictable nature are presumed to be accomplished by some additional facility not connected with the problem.

The company possesses a single-channel repair facility (i.e., only one machine can be repaired at a time). Each repair job is assigned a numerical priority index p , which in Cobham's formulation takes on integral values ranging between unity and a positive integer r . A job with priority p_1 is worked on before any with priority p_2 , if $p_1 < p_2$. Jobs of equal priority are worked on in the order of their arrival. Jobs of priority p arrive in Poisson law fashion at an average rate λ_p ; evidently, $\lambda = \sum_1^r \lambda_p$. The average arrival rate of jobs actually worked on by the maintenance facility is

$$\lambda^* = \sum_1^r f_p \lambda_p, \quad (1)$$

where f_p is the probability that a job of priority p receives service from the maintenance facility. This quantity is introduced in the interest of unifying Cobham's treatments of the saturated⁶ and unsaturated⁴ cases, defined below; it is evaluated in a later paragraph.

-
- ¹ Reproduced with written permission of the Operations Research Society of America. It has been taken from:
Phipps T. E. "Machine Repair as a priority waiting-line problem." Operations Research. 4 (January 1956), pp. 76-86
- ² Cobham A. Journal of Operations Research Society, 3, 1955. p. 547

The probability that a job of priority p requires a repair time less than t is denoted by $F_p(t)$. The average cumulative job repair time distribution for units of all priorities is

$$F(t) = \frac{1}{\lambda} \sum_1^r \lambda_p F_p(t).$$

The corresponding repair time distribution for units actually worked on by the repair facility is

$$F^*(t) = \frac{1}{\lambda^*} \sum_1^r f_p \lambda_p F_p(t). \quad (2)$$

Of course, $\int_0^\infty dF_p(t) = \int_0^\infty dF(t) = \int_0^\infty dF^*(t) = 1$.

The average repair time for units of the p th priority level is

$$\frac{1}{\mu_p} = \int_0^\infty t dF_p(t). \quad (3)$$

The average repair rate μ^* for jobs worked on by the facility is seen to be

$$\frac{1}{\mu^*} = \left(\sum_1^r \frac{f_p \lambda_p}{\mu_p} \right) / \left(\sum_1^r f_p \lambda_p \right). \quad (4)$$

If the repair facility is adequate to handle all work required of it, $f_p < 1$ for all p , and we shall speak of the system as 'unsaturated,' or as operating 'below saturation.' In this case the average repair rate μ is given by

$$\frac{1}{\mu} = \frac{1}{\mu^*} = \frac{1}{\lambda} \sum_1^r \frac{\lambda_p}{\mu_p}. \quad (\text{Unsaturated case})$$

Since the repair rate capability must exceed the job arrival rate for unsaturated activity, i.e., $\mu > \lambda$, the condition for the obtaining of this case is

$$\sum_1^r \frac{\lambda_p}{\mu_p} < 1. \quad (\text{Unsaturated case}) \quad (5)$$

If this inequality is not satisfied, $f_p < 1$ for some p , and the system will be said to be 'saturated,' implying the growth of an infinite waiting line.

$$\sum_1^r \frac{\lambda_p}{\mu_p} \geq 1. \quad (\text{Saturated case}) \quad (6)$$

In this case the average repair rate of jobs admitted to the maintenance facility must equal the average arrival rate of such jobs, $\mu^* = \lambda^*$. From equations (1) and (4), this implies that

$$\sum_1^r \frac{f_p \lambda_p}{\mu_p} = 1, \quad (\text{Saturated case}) \quad (7)$$

at a steady state. This relation provides a determining condition on the f_p together with the evident requirements that $0 \leq f_p \leq 1$ for all p , and that no f_{p_1} can be less than unity if any $f_{p_2} > 0$ for $p_2 > p_1$. (That is, if any low priority units are worked on, all higher priority units must certainly be serviced.) Thus, in the saturated case equation (7) determines an integer s and a fraction m , $0 < m \leq 1$, such that

$$f_p = \begin{cases} 1 & \text{for } p < s, \\ m & \text{for } p = s, \\ 0 & \text{for } p > s. \end{cases} \quad \text{(Saturated case)} \quad (8)$$

As previously stated, in the alternative case,

$$f_p = 1 \quad \text{for all } p. \quad \text{(Unsaturated case)} \quad (9)$$

The determining equation for s and m is

$$\sum_{p=1}^{s-1} \frac{\lambda_p}{\mu_p} + m \frac{\lambda_s}{\mu_s} = 1. \quad (10)$$

Cobham's final result⁴ for the expected waiting time W_p of a job of priority p , expressed in his equation (3), may be written

$$W_p = W_0 / \left[\left(1 - \sum_{k=1}^{p-1} \frac{f_k \lambda_k}{\mu_k} \right) \left(1 - \sum_{k=p}^{\infty} \frac{f_k \lambda_k}{\mu_k} \right) \right]. \quad (11)$$

We have taken the liberty of inserting f -factors in the denominator for consistency with the rest of our development. These alter nothing in the unsaturated case and provide a necessary correction⁶ in the saturated case. The constant W_0 is evaluated⁶ as

$$W_0 = \frac{1}{2} \lambda^* \int_0^{\infty} t^2 dF^*(t). \quad (12)$$

It is convenient to introduce the concept of an *active waiting line*, each member of which is certain to receive eventual service from the facility. The expected number of jobs in the active waiting line is

$$N = \sum_{p=1}^{\infty} f_p \lambda_p W_p. \quad (13)$$

In contrast, saturated activity produces a category of low-priority units that never receive service from the facility in question. These units we shall call *permanent duds*, by analogy with the case of aircraft repair, in which aircraft too seriously damaged for local repair are sometimes so termed. Permanent duds will be thought of as distinct from jobs entering the active waiting line, and subject to altogether different treatment.

The critical job duration τ , analogous to the critical priority level, determined by the means discussed below.

The analogue of $1/\mu_p$, the average servicing time for units of priority level is the average repair time for jobs of duration t , which, of course, is just equal to t . Similarly, $F_{t'}(t)$, the analogue of $\text{Cob}(t)$, $F_p(t)$, is the cumulative job repair time distribution for units of priority level t . That is, it is the probability that a job of duration t' will have a repair time less than t . As such, it is a step function,

$$F_{t'}(t) = \begin{cases} 0 & \text{for } t' > t, \\ 1 & \text{for } t' \leq t. \end{cases} \quad (1)$$

The continuous analogue of equation (1), above, is

$$\lambda^* = \int_0^\infty f(t) \lambda_t dt = \lambda \int_0^\tau dF(t) = \lambda F(\tau). \quad (2)$$

The analogue of equation (2) is

$$\begin{aligned} F^*(t) &= \frac{1}{\lambda^*} \int_0^\infty f(t') \lambda_{t'} dt' F_{t'}(t) \\ &= \frac{1}{\lambda^*} \int_0^t f(t') \lambda_{t'} dt' = \frac{\lambda}{\lambda^*} \int_0^t f(t') dF(t'). \end{aligned} \quad (3)$$

For $t \leq \tau$, this implies that

$$F^*(t) = F(t)/F(\tau). \quad (4)$$

The analogue of equation (3) is

$$\begin{aligned} \frac{1}{\mu^*} &= \left(\int_0^\infty f(t) \lambda_t dt \cdot t \right) / \left(\int_0^\infty f(t) \lambda_t dt \right) = \left(\int_0^\tau t dF(t) \right) / \left(\int_0^\tau dF(t) \right) \\ &= \frac{1}{F(\tau)} \int_0^\tau t dF(t). \end{aligned} \quad (5)$$

The case of non-saturation is obtained from these formulas by replacing τ by ∞ and noting that $F(\infty) = 1$.

Conditions defining the saturated and unsaturated cases are obtained by analogy with equations (5) and (6), viz.,

$$\int_0^\infty \lambda_t dt \cdot t = \lambda \int_0^\infty t dF(t) < 1, \quad (\text{Unsaturated case}) \quad (6)$$

$$\text{and} \quad \lambda \int_0^\infty t dF(t) \geq 1. \quad (\text{Saturated case}) \quad (7)$$

The critical job duration τ is determined by analogy with equation (10), viz.,

$$\int_0^{\infty} f(t) \lambda_t dt \cdot t = \int_0^{\tau} \lambda_t dt \cdot t = \lambda \int_0^{\tau} t dF(t) = 1. \quad (22)$$

If $f(t)$ behaves in such a way that ambiguity arises, it is to be understood that τ has the *smallest* value satisfying equation (22). (In fact, there are certain mathematical restrictions on $dF(t)$ that we have not stated. For practical purposes it is simplest to rule out infinite singularities of this function, although the formal treatment could be modified to admit a plurality of such.)

From equations (16), (19), and (22) the reader will readily confirm that $\mu^* = \lambda^*$ in the case of saturation.

Both summations appearing in the denominator of equation (11) become in the continuous limit

$$\sum_1 \frac{f_k \lambda_k}{\mu_k} \rightarrow \int_0^t f(t') \lambda_{t'} dt' \cdot t' = \lambda \int_0^t f(t') t' dF(t') = \lambda \int_0^t t' dF(t') \quad \text{for } t \leq \tau.$$

Consequently our final result for the expected waiting time of a job of duration t is

$$W_t = W_0 / \left(1 - \lambda \int_0^t t' dF(t') \right)^2 \quad \text{for } t \leq \tau$$

$$W_t = \infty \quad \text{for } t > \tau. \quad (23)$$

Making use of the relation $\lambda^* dF^*(t) = f(t) \lambda dF(t)$, which follows from equation (17), we evaluate the constant W_0 [cf. equation (12)] as

$$W_0 = \frac{1}{2} \lambda \int_0^{\infty} f(t) t^2 dF(t) = \frac{1}{2} \lambda \int_0^{\tau} t^2 dF(t). \quad (24)$$

The active waiting line consists of jobs of duration less than τ ; all other jobs are permanent duds. The analogue of equation (13) is

$$N = \int_0^{\infty} f(t) \lambda_t dt W_t = W_0 \int_0^{\tau} \left[dF(t) / \left(1 - \lambda \int_0^t t' dF(t') \right)^2 \right]. \quad (25)$$

This integral is improper for the case of saturation, so that the active waiting line, as before, is infinite. It acquires this character of infiniteness as an accumulation of job durations having τ as a least upper bound. Thus, for any $\epsilon > 0$ the number of jobs of duration less than or equal to $\tau - \epsilon$ in the active waiting line is finite. In the case of non-saturation the integral in equation (25) is finite, so that there is a finite waiting line. Other qualitative features of the discrete priority index problem carry

over to the continuous formulation. The greater formal simplicity of the latter will be noted.

It is of interest to apply the general formulas just derived in a specific example.

Exponential Repair Times

The cumulative distribution function of repair times is $F(t) = 1 - e^{-\mu t}$. From equations (20) and (21) it is seen that saturation occurs if $\lambda \geq \mu$.

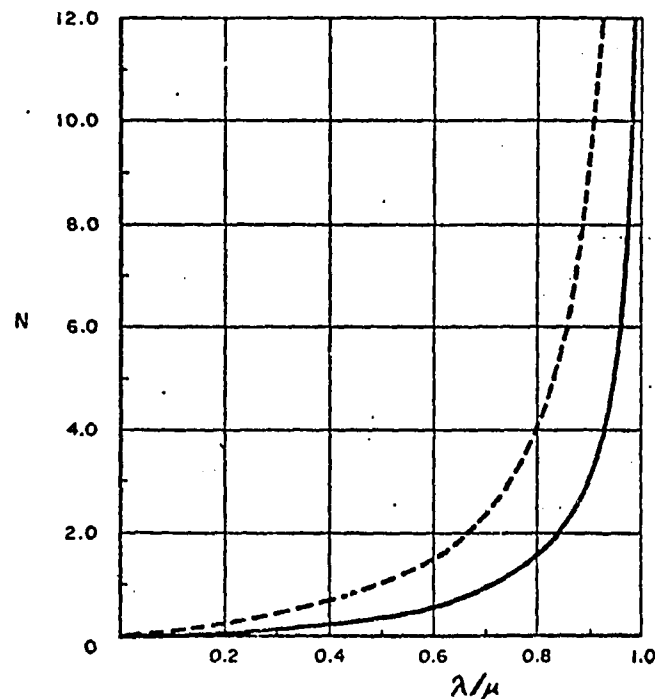


Fig. 1. Expected number N of units in the waiting line, as a function of 'utilization factor,' λ/μ , for exponential servicing times, unsaturated system. The dotted curve is the conventional result for jobs treated on a 'first come, first served' basis; the solid curve applies to the case in which shortest jobs are given highest priority.

nonsaturation if $\lambda < \mu$. From equation (22) the critical job duration in the case of saturation is given by the solution of $(\lambda/\mu) [1 - e^{-\mu\tau}(1 + \mu\tau)] = 1$. In this case the average repair rate μ^* of jobs in the active waiting line is given, with the help of equation (16), by $\mu^* = \lambda^* = \lambda F(\tau) = \lambda (1 - e^{-\mu\tau})$. In the case of non-saturation $\mu^* = \mu$.

The constant W_0 , given by equation (24), is evaluated to be

$$W_0 = \lambda/\mu^2, \quad (\text{Unsaturated case})$$

$$W_0 = (1/\mu)[1 - \lambda\tau e^{-\mu\tau}]. \quad (\text{Saturated case})$$

From equation (23) the expected waiting time for a job of duration t is

$$W_t = W_0 / \{1 - (\lambda/\mu)[1 - e^{-\mu t}(1 + \mu t)]\}^2.$$

In the saturated case it is necessary to make the stipulation that $t \leq \tau$ in the expression just given; for t -values greater than τ , W_t is infinite.

The expected number of jobs in the waiting line in the unsaturated case is given by equation (25),

$$N = \frac{\lambda^2}{\mu} \int_0^{\infty} \frac{e^{-\mu t} dt}{\{1 - (\lambda/\mu)[1 - e^{-\mu t}(1 + \mu t)]\}^2}.$$

N is infinite in the saturated case.

The quantity N , calculated numerically from this formula, is graphed in Fig. 1 (solid curve). Also shown (dotted curve) is the expected length of the waiting line for priority assignment on a 'first come, first served' basis. The latter is given by the well-known formula, $N = (\lambda/\mu) / [1 - (\lambda/\mu)]$. Exponential holding times and an unsaturated system are assumed. Comparison of the two curves shows the effectiveness of an optimum priority system in reducing average waiting line lengths.

APPENDIX THIRTEEN

Proof of the Response Time Theorem
 FB_N model (eq. 7.1.4)

As noted, a single server is in charge of N customers, and the time elapsing from the moment a customer is dismissed until his next arrival at the queue is exponentially distributed with a mean of $1/\lambda$. Service times are assumed to be mutually independent and identically distributed random variables having an exponential density with mean $1/\mu$. The quantum setting-up and processing times are assumed to be constants and denoted by τ and θ , respectively. Each time a customer is admitted, he is given at most θ time units of service. If the service is not completed within this time interval θ , the customer is dismissed to the end of the queue. Otherwise, he departs at the instant of service completion, and a new quantum may be started. It is assumed that setting-up requiring τ time units takes place at the beginning of each quantum.

Denoting the length of a quantum by Q , we have:

$$Q \stackrel{d}{=} \begin{cases} \tau + \theta & \text{with probability } \alpha = e^{-\mu\theta}, \\ D & \text{with probability } 1 - \alpha, \end{cases} \quad (1)$$

$\stackrel{d}{=}$ denoting equal in distribution, and D having a density $f_D(\cdot)$.

$$f_D(x) = \mu e^{-\mu(x-\tau)} / (1 - \alpha), \quad \tau \leq x \leq \theta + \tau. \quad (2)$$

The expectation of the quantum's length is easily derived as

$$E(Q) = \tau + (1 - \alpha)/\mu. \quad (3)$$

The total time devoted by the server to a single customer, including setting-up times, is denoted by S :

$$S = (R - 1)(\tau + \theta) + D, \quad (4)$$

where R is the number of service quanta given to the customer and is geometrically distributed:

$$P(R = r) = \alpha^{r-1}(1 - \alpha), \quad r = 1, 2, \dots \quad (5)$$

The Laplace-Stieltjes (L.S.) transform of a nonnegative random variable X with a distribution function $F_X(\cdot)$ is denoted here by $L_X(z)$:

$$L_X(z) = \int_0^\infty e^{-zx} dF_X(x). \quad (6)$$

Since R and D are mutually independent, we can easily derive the L.S. transform of S using eqs. (4) and (5).

$$L_S(z) = \sum_{r=1}^{\infty} \alpha^{r-1}(1 - \alpha) e^{-z(\theta+\tau)(r-1)} L_D(z) = \frac{\mu e^{-z\tau}(1 - \alpha e^{-\theta z})}{(\mu + z)(1 - \alpha e^{-z(\theta+\tau)})}. \quad (7)$$

The first moment of S is

$$E(S) = 1/\mu + \tau/(1 - \alpha). \quad (8)$$

The proof is reproduced here with the written permission of the Association from Computing Machinery. It has been taken from I. Adiri and E. Ivi-Itzhak. "A Time-Sharing Queue with Finite Number of Customers." Journal of ACM, Vol. 10, No. 2, April 1969 p. 301.

From eq. (5) we obtain

$$E(R) = 1/(1 - \alpha). \quad (9)$$

Steady-State Number of Customers in the System at Epochs of Departure. Let X_n be the number of customers in the system (in the queue and in service) immediately after the n th departure. The time elapsing between the n th and the $(n + 1)$ -th departures is distributed as S provided $X_n > 0$. If $X_n = 0$, we are concerned with the time interval (also distributed as S) between the $(n + 1)$ -th arrival and the $(n + 1)$ -th departure, hence

$$X_{n+1} = X_n + Y_n - 1, \quad (10)$$

where Y_n is the number of arrivals between the n th and $(n + 1)$ -th departures. It is easy to show that $\{X_n\}$ is a Markov chain and is statistically the same as the Markov chain encountered in the classical machine-interference model with service times distributed as S and the breakdown intensity of each of the N machines being equal to λ . This machine-interference model was investigated by Khintchine,² Kronig,³ and Kronig and Mondria,⁴ and much later by Takacs

Let π_j , $j = 0, 1, 2, \dots, N - 1$, be the steady-state probability of there being j customers in the system immediately after an epoch of departure. Takacs² shows that

$$\pi_j = \sum_{r=N-j-1}^{N-1} (-1)^{N-j-1-r} \binom{N-r-1}{j} B_r, \quad (11)$$

where

$$B_r = C_r \frac{\sum_{i=0}^{N-1} \binom{N-1}{i} \frac{1}{C_i}}{\sum_{i=0}^{N-1} \binom{N-1}{i} \frac{1}{C_i}}, \quad (12)$$

and

$$C_r = \begin{cases} 1 & \text{for } r = 0, \\ \prod_{i=1}^r L_S(i\lambda)/(1 - L_S(i\lambda)) & \text{for } r = 1, 2, \dots, N - 1. \end{cases} \quad (13)$$

Expected Total Response Time. The total response time T is defined as the steady-state time from arrival to departure which a customer spends in the system. To derive $E(T)$ we use a simple argument suggested by Khintchine [5], namely, that the expected number of arrivals per unit time must equal the expected number of departures per unit time. Since each customer undergoes a cycle of expected length $E(T) + 1/\lambda$, the expected number of arrivals per unit time equals $N/(E(T) + 1/\lambda)$. On the other hand, when the server is busy, the expected departure rate equals $1/E(S)$. Therefore the expected number of departures per unit time equals $(1 - P_0)/E(S)$ where P_0 is the steady-state probability of the server being idle. Hence

$$(1 - P_0)/E(S) = N/(E(T) + 1/\lambda). \quad (14)$$

² Lajos Takacs. Introduction to Theory of Queues. (New York, Oxford, University Press), p. 194.

which yields

$$E(T) = NE(S)/(1 - P_0) - 1/\lambda. \quad (15)$$

The expected number of customers served in one busy period equals $1/\pi_0$. (Note that π_0 is the "long-run" proportion of served customers who terminate busy periods. In other words, on the average, one out of every $1/\pi_0$ customers terminates a busy period.) Therefore a busy period is of average length $E(S)/\pi_0$. The expected length of an idle period is $1/(N\lambda)$. Since P_0 is the fraction of time during which the server is idle, we may write

$$P_0 = \frac{1/(N\lambda)}{1/(N\lambda) + E(S)/\pi_0} = \frac{\pi_0}{\pi_0 + N\lambda E(S)}, \quad (16)$$

where π_0 is obtainable from eqs. (11) and (12) as

$$\pi_0 = \frac{1}{\sum_{i=0}^{N-1} \binom{N-1}{i} \frac{1}{C_i}}, \quad (17)$$

and C_i is given by eq. (13).

Expected Total Response Time Conditional upon Length of Service Demand. In a time-sharing system, the expected total response time of a customer is an increasing function of the length of service demanded by him. This is one of the main advantages of the time-sharing system, and therefore it is important to derive an expression for the fundamental relation between expected total response time and length of service demand.

We define the i th stationary response time of a customer, denoted by T_i , $i = 2, 3, \dots$, as the time elapsing between his $(i-1)$ -th and i th admissions, assuming that he demands at least i service quanta. Similarly, the first response time, T_1 , is the time elapsing from the moment of the customer's arrival until his first admission. $E(T | L = l)$ denotes the expected stationary total response time of a customer whose service demand is of length l . Clearly

$$E(T | L = l) = \sum_{i=1}^{\lceil \frac{l}{\theta} \rceil} E(T_i) + l - \theta \left(\left\lceil \frac{l}{\theta} \right\rceil - 1 \right) + \tau, \quad (18)$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x .

Let K_i , $i = 1, 2, \dots$, denote the steady-state number of customers in the queue behind a customer just admitted for the i th time. The i th response time, $i > 1$, is strongly dependent on K_{i-1} :

$$\begin{aligned} E(T_i | K_{i-1} = k) \\ = \tau + \theta + E(Q) \left(k + \sum_{j=1}^{N-k-1} j \binom{N-k-1}{j} (1-\beta)^j \beta^{N-k-j-1} \right) \\ = \tau + \theta + E(Q)(k + (1-\beta)(N-k-1)) \end{aligned} \quad (19)$$

where $\beta = e^{-\lambda(\theta+\tau)}$.

From eq. (19) we obtain the expectation of T_i as

$$E(T_i) = \tau + \theta + E(Q)(1-\beta)(N-1) + E(Q)\beta E(K_{i-1}). \quad (20)$$

Summing both sides of eq. (20) with respect to i , we obtain

$$\sum_{i=1}^{\left[\frac{l}{\theta}\right]} E(T_i) = \left(\left[\frac{l}{\theta}\right] - 1\right) (\tau + \theta + E(Q)(1 - \beta)(N - 1)) + E(Q)\beta \sum_{i=1}^{\left[\frac{l}{\theta}\right]-1} E(K_i), \quad (21)$$

and substituting in eq. (18),

$$E(T | L = l) = E(T_1) + \left(\left[\frac{l}{\theta}\right] - 1\right) (\tau + \theta + E(Q)(1 - \beta)(N - 1)) + E(Q)\beta \sum_{i=1}^{\left[\frac{l}{\theta}\right]-1} E(K_i) + l - \theta \left(\left[\frac{l}{\theta}\right] - 1\right) + \tau. \quad (22)$$

In this expression we still have to determine $\sum_{i=1}^{\left[\frac{l}{\theta}\right]-1} E(K_i)$ and $E(T_i)$.

Let $p_{i,j}$, $i, j = 0, 1, \dots, N$, denote the probability of there being j customers in the queue at the instant of a quantum completion given that at the instant of the previous quantum completion there had been i customers in the queue. These transition probabilities were determined in [2].

$$p_{ij} = \begin{cases} 0, & j < i - 1, \\ \int_0^\infty \mu e^{-\mu x} e^{-\lambda(x+\tau)} dx, & j = i - 1, \\ \mu e^{-\mu x} \binom{N-i}{j-i+1} (1 - e^{-\lambda(x+\tau)})^{j-i+1} (e^{-\lambda(x+\tau)})^{N-j-1} dx \\ + e^{-\theta\mu} \binom{N-i}{j-i} (1 - e^{-\lambda(\theta+\tau)})^{j-i} (e^{-\lambda(\theta+\tau)})^{N-j}, & j \geq i, \end{cases} \quad (23)$$

where $i \geq 1$, $p_{0,j} = p_{1,j}$ for all j , and

$$\binom{m}{n} = \begin{cases} \frac{m!}{n!(m-n)!} & \text{if } m \geq n, \\ 0 & \text{if } n > m. \end{cases} \quad (24)$$

Taking the integral and rearranging terms in eq. (23) yields

$$p_{ij} = \begin{cases} 0, & j < i - 1, \\ \frac{\mu e^{-\lambda\tau}}{\mu + \lambda} (1 - e^{-\theta(\mu+\lambda)}), & j = i - 1, \\ \binom{N-i}{j-i+1} e^{-\lambda\tau(N-j-1)} \\ \cdot \sum_{r=0}^{j-i+1} \frac{\binom{j-i+1}{r} (-1)^r e^{-\lambda\tau r} (1 - e^{-\theta(\mu+\lambda(N-j-1+r))})}{\mu + \lambda(N-j-1+r)} \\ + e^{-\theta\mu} \binom{N-1}{j-1} (1 - e^{-\lambda(\theta+\tau)})^{j-i} (e^{-\lambda(\theta+\tau)})^{N-j}, & j \geq i. \end{cases} \quad (25)$$

A straightforward argument yields the following relation:

$$P(K_i = k | K_{i-1} = j) = p_{j,k}^* = \sum_{r=0}^{N-j-1} \binom{N-j-1}{r} (1 - e^{-\lambda(\theta+r)})^r \cdot (e^{-\lambda(\theta+r)})^{N-j-1-r} p_{j+1+r,k+1}^{(j+r)}, \quad j, k = 0, 1, \dots, N-1, \quad (26)$$

where $p_{i,j}^{(n)}$ is the (i, j) -th element of the matrix $\{p_{i,j}^{(n)}\}$. A matrix to the power 0 is defined in this paper as equal to the unit matrix.

In a similar fashion, we have

$$P(K_i = k | K_i = j) = p_{j,k}^{*(i-1)}, \quad i = 1, 2, 3, \dots, \quad (27)$$

where $p_{j,k}^{*(n)}$ is the (j, k) -th element of the matrix $\{p_{j,k}^{*(n)}\}$.

Multiplying eq. (27) by $P(K_i = j)$ and summing over all j , we obtain

$$P(K_i = k) = \sum_{j=0}^{N-1} p_{j,k}^{*(i-1)} P(K_i = j), \quad i = 1, 2, 3, \dots \quad (28)$$

Note that the random variables K_i , $i = 1, 2, \dots$, are not identically distributed. If the distribution of K_1 is given, it is possible to calculate that of K_i for all i . Krishnamoorthi and Wood³ assume that K_1 is distributed as $K - 1$ where K is the steady-state number of customers in the system immediately after the beginning of a quantum selected at random. This is not generally true and is the main discrepancy between their partial results and ours. Obviously the following relation must hold:

$$P(K - 1 = k) = \sum_{i=1}^{\infty} P(K_i = k) P(R \geq i) / E(R), \quad (29)$$

$$K = 0, 1, 2, \dots, N - 1.$$

The argument behind it being that $P(R \geq i) / E(R)$ is the probability that a randomly selected quantum is the i th for some customer.

Substitution of eqs. (28) and (5) yields

$$P(K - 1 = k) = (1 - \alpha) \sum_{i=1}^{\infty} \sum_{j=0}^{N-1} p_{j,k}^{*(i-1)} P(K_i = j) \alpha^{i-1}, \quad (30)$$

$$k = 0, 1, 2, \dots, N - 1.$$

The distribution of K is obtainable from eq. (23) and is given in [2]. To determine $P(K_i = j)$, $j = 0, 1, \dots, N - 1$, we must solve N linear equations given by eq. (30). After solving and substituting the results in eq. (28), we obtain $E(K_i)$, $i = 1, 2, \dots$. In turn, we substitute these in eq. (22) and obtain the relation linking $E(T | L = l)$ to $E(T_1)$, as well as some other terms given as functions of l and the known parameters of the system. Noting that $E(T)$ has been determined in eq. (15), we multiply both sides of eq. (22) by $\mu e^{-\mu l}$ and integrate over all nonnegative values of l , and thus obtain a relation between $E(T)$ and $E(T_1)$.

$$\begin{aligned} E(T) &= \int_0^{\infty} E(T | L = l) \mu e^{-\mu l} dl, \\ &= E(T_1) + \tau + 1/\mu + (\tau + E(Q)(1 - \beta)(N - 1))\alpha / (1 - \alpha) \\ &\quad + E(Q)\beta \sum_{i=1}^{\infty} \alpha^i E(K_i). \quad (31) \end{aligned}$$

³ B. Krishnamoorthi and R. Wood. "Time-Shared Computer with both Inter-arrival and service times exponential." Journal of ACM, Vol. 13, No. 3, (July 1966), pp. 317-338

Combining eqs. (22) and (31) yields

$$\begin{aligned}
 E(T|L=l) = E(T) - 1/\mu + l \\
 + (\tau + E(Q)(1-\beta)(N-1)) \left(\left[\frac{l}{\theta} \right] - 1/(1-\alpha) \right) \\
 + E(Q)\beta \left(\sum_{i=1}^{\left[\frac{l}{\theta} \right]-1} E(K_i) - \sum_{i=1}^{\infty} \alpha^i E(K_i) \right). \quad (32)
 \end{aligned}$$

APPENDIX FOURTEEN
 Proof of The Response time Theorems¹
 Feedback Model
 (Eq. 7.1.16)

MATHEMATICAL MODEL: RR, ²

CUSTOMERS ARRIVE at a single-server facility according to a homogeneous Poisson process with intensity λ . Service times (as demanded by customers) are independent random variables identically distributed exponentially with mean $1/\mu$. The waiting line consists of r separate queues. A newly arrived customer joins the end of the first queue; upon admittance to service he is eligible for θ_1 units of service time. If his service demand is shorter than θ_1 , he completes his service and leaves the system; otherwise he receives θ_1 units of service time and joins the end of the second queue. In general, at the i th queue ($i=1, 2, \dots, r-1$) the customer is eligible for θ_i units of service time during which he may complete his service and leave the system, or join the end of the $(i+1)$ th queue after receiving θ_i units of service time. A customer in the r th queue who does not complete his service within θ_r service time units rejoins the end of the same queue. Having completed a quantum at any one of the queues, the server admits to service the customer who is first in the lowest-index nonempty queue. We assume that each quantum starts with a set-up time due to swapping, housekeeping, etc. The i th quantum thus comprises two elements: a set-up time of length τ_i and a processing time not exceeding θ_i units.

Busy Period and Backlog

Let

$$\alpha_i = e^{-\mu\theta_i}, \quad (1)$$

$$\beta_i = \exp(-\mu \sum_{j=1}^{i-1} \theta_j) = \prod_{j=1}^{i-1} \alpha_j, \quad (2)$$

$$q_i = \sum_{j=1}^{i-1} (\theta_j + \tau_j), \quad (i=1, 2, \dots) \quad (3)$$

¹ Reproduced with the written permission of the Operations Research Society of America. It has been taken from:

I. Adiri and B. Avi-Itzhak. "A Time-Sharing with Many Queues." *Operations Research*. Vol. 17, No. 6, (November-December 1969), pp. 1077-1089.

² In the dissertation the model RRr has been referred to as FBr

where $(1-\alpha_i)$ is the probability of service being completed in the i th queue, $i \leq r$. For $i > r$, θ_i and τ_i are defined as equal to θ_r and τ_r , respectively. Hence, for $i > r$ we have $\alpha_i = \alpha_r$ and $q_i = q_r + (i-r)(\theta_r + \tau_r)$.

The length of a quantum devoted to a customer in the i th queue ($i=1, 2, \dots, r$) behaves as a random variable that we shall denote by Q_i .

$$Q_i \stackrel{d}{=} \begin{cases} Q_i + \tau_i & \text{with probability } \alpha_i, \\ D_i & \text{with probability } 1 - \alpha_i, \end{cases} \quad (4)$$

where D_i is an r.v. possessing the density

$$f_{D_i}(x) = \mu e^{-\mu(x-\tau_i)} / (1-\alpha_i), \quad \tau_i \leq x \leq \theta_i + \tau_i, \quad (i=1, 2, \dots, r) \quad (5)$$

For $i > r$, $Q_i \stackrel{\text{def}}{=} Q_r$ and $D_i \stackrel{\text{def}}{=} D_r$.

Within each queue, quantum lengths are mutually independent and also independent of the number of customers present in the various queues. This property is general for all service-time distributions, except for the r th queue, where it may not hold for nonexponential service times. The quanta in the different queues are mutually independent. Let the r.v. S denote the total time (including set-ups) devoted by the server to a customer:

$$S \stackrel{d}{=} \begin{cases} D_1 & \text{with probability } 1 - \alpha_1, \\ q_i + D_{i+1} & \text{with probability } \beta_i(1 - \alpha_{i+1}). \end{cases} \quad (i=1, 2, \dots) \quad (6)$$

The backlog and the busy period of the RR r model behave as in a $M/G/1$ model with service times distributed as S . This equivalence holds also for RR 1 systems with general time distributions, where S is defined as the total time devoted to a customer.

Total Response Time

The total response time of a customer is defined as the total time he spends in the system from arrival to departure.

Let $X_n (n=1, 2, \dots)$ denote the number of customers in the RR r system at the moment of the n th departure. Consider now a similar RR ∞ model with the same system parameters except that for $i > r$, $\theta_i = \theta_r$ and $\tau_i = \tau_r$, and let X'_n denote the number of customers in the RR ∞ system at the moment of the n th departure. Clearly, X_n and X'_n are equal in distribution. This is a result of the premise that service times are exponentially distributed. Note that the two systems are identical except for customers who have completed at least $r-1$ quanta and are still present in the system. These customers are subject to one queue discipline in the RR r system and to another queue discipline in the RR ∞ system. According to the assumptions that for $i > r$, $\theta_i = \theta_r$ and $\tau_i = \tau_r$.

and that service times are exponentially distributed, the order of service in the r th queue does not

queue. By LITTLE's theorem, the response time [denoted by l]

Using Schrage's results^[1] where n is an integer such that

$$E(T) = \int_0^{\infty} E(T|L=l) dl \quad \sum_{i=1}^{n-1} \theta_i < l \leq \sum_{i=1}^n \theta_i.$$

$$= \frac{1}{\mu} + \sum_{n=1}^{\infty} \beta_{n-1} (1 - \alpha_n) \left\{ \frac{\lambda E(U_n^2) + \lambda \sum_{i=n+1}^{\infty} \beta_{i-1} E(Q_i^2)}{2[1 - \lambda E(U_{n-1})][1 - \lambda E(U_n)]} + \frac{q_{n-1}}{1 - \lambda E(U_{n-1})} - \sum_{i=1}^{n-1} \theta_i + \tau_n \right\}. \quad (7)$$

In this expression $E(T|L=l)$ is the total response-time expectation of a customer who demands l service-time units in the RR ∞ model, $\alpha_0 = \beta_0 = 1$, $\rho_0 = \theta_0 = \tau_0 = 0$, and

$$U_n^d = \begin{cases} q_{i-1} + D_i & \text{with probability } \beta_{i-1}(1 - \alpha_i), \quad (i=1, 2, \dots, n) \\ q_n & \text{with probability } \beta_n, \quad (n=1, 2, \dots) \end{cases} \quad (8)$$

$$U_n^d = \begin{cases} q_n \beta_n + \sum_{i=1}^{n-1} \beta_{i-1} (1 - \alpha_i) [q_{i-1} + E(D_i)], \\ q_n^2 \beta_n + \sum_{i=1}^{n-1} \beta_{i-1} (1 - \alpha_i) [q_{i-1}^2 + 2q_{i-1} E(D_i) + E(D_i^2)]. \end{cases} \quad (9)$$

($n=1, 2, \dots$)

Conditional Total Response Time: RR r

The total steady-state response time of a customer with service requirement l is denoted by $T(l)$. This quantity will be referred to as the conditional total-response time.

The behavior of the first $r-1$ queues is the same in both the RR r and the RR ∞ systems. Thus for $l \leq \sum_{i=1}^{r-1} \theta_i$ we may use Schrage's results^[2]:

$$E[T(l)] = [\lambda E(U_n^2) + \lambda \sum_{i=n+1}^{\infty} \beta_{i-1} E(Q_i^2)] / \{ 2[1 - \lambda E(U_{n-1})][1 - \lambda E(U_n)] + q_{n-1}/[1 - \lambda E(U_{n-1})] + l - \sum_{i=1}^{n-1} \theta_i + \tau_n \}, \quad (10)$$

where n is an integer such that

$$\sum_{i=1}^{n-1} \theta_i < l \leq \sum_{i=1}^n \theta_i. \quad (11)$$

Customers with service time $l > \sum_{i=1}^{r-1} \theta_i$ reach the r th queue, where they circulate until departure. We define the i th response time ($i=2, 3, \dots$) of a customer (denoted by T_i) as the time elapsing between his $(i-1)$ th and i th admissions to service, assuming that $l > \sum_{j=1}^{i-1} \theta_j$. Similarly, T_1

is the time elapsing between his arrival and first admission. Let K_i+1 denote the number of customers present in the i th queue at the time of the i th admission to service. For $i=r, r+1, \dots$, K_i+1 is the number in the r th queue at the moment of an i th admission. The i th response time expectation $E(T_i)$, $i=r+1, r+2, \dots$, is obtained as:

$$E(T_i) = [\theta_r + \tau_r + E(K_{i-1})E(Q_r)]/[1 - \lambda E(U_{r-1})], \quad (12)$$

$$(i=r+1, r+2, \dots)$$

where $r \geq 2$.

The argument supporting (12) is that T_i may be viewed as a busy period starting with initial expected backlog of magnitude $[\theta_r + \tau_r + E(K_{i-1})E(Q_r)]$ in a $M/G/1$ system where the arrival rate is λ and service times are distributed as U_{r-1} .

For $\sum_{i=1}^{n-1} \theta_i < l \leq \sum_{i=1}^n \theta_i$ we have:

$$E[T(l)] = \sum_{i=1}^{n-1} E(T_i) + l - \sum_{i=1}^{n-1} \theta_i + \tau_n. \quad (13)$$

Substitution of (12) yields, for $n \geq r$,

$$E[T(l)] = \sum_{i=1}^{n-1} E(T_i) + [(n-r)(\theta_r + \tau_r) + E(Q_r) \sum_{i=r}^{n-1} E(K_i)] / [1 - \lambda E(U_{r-1})] + l - \sum_{i=1}^{n-1} \theta_i + \tau_n. \quad (14)$$

In this expression, both sums are unknown. We should like to express them as functions of $E(K_r)$. The following recursive relation is readily shown to hold for $i > r$,

$$E(K_i) = E(T_i)\lambda\beta_{r-1} + E(K_{i-1})\alpha_r. \quad (15)$$

From (12) and (15) we obtain:

$$\sum_{i=r}^{n-1} E(K_i) = [E(K_r) - a/(1-b)](1-b^{n-r})/(1-b) + (n-r)a/(1-b), \quad (16)$$

where

$$a = \lambda\beta_{r-1}(\theta_r + \tau_r)/[1 - \lambda E(U_{r-1})] \quad (17)$$

and

$$b = \alpha_r + \lambda\beta_{r-1}E(Q_r)/[1 - \lambda E(U_{r-1})]. \quad (18)$$

Let T_e be the time elapsing between the arrival of a customer whose service time exceeds $\sum_{i=1}^{n-1} \theta_i$ and the next admission of a customer in the r th queue and let Z denote the number of customers in the r th queue when a new customer joins it. The time elapsing between admission of a customer from the r th queue until the server is again ready to admit a customer from it is distributed as a busy period of an $M/G/1$ model, where service times are distributed as U_{r-1} and the starting backlog is distributed as Q_r . Bearing in mind the definitions of T_e and Z , we thus have

$$\sum_{i=1}^{\infty} E(T_i) = E(T_e) + [E(Z)E(Q_r)]/[1 - \lambda E(U_{r-1})]. \quad (19)$$

The expectation of T_e is obtained by using Schrage's^[2] approach:

$$E(T_e) = [\lambda E(U_{r-1}^2) + \lambda E(Q_r^2)\beta_{r-1}/(1 - \alpha_r)]/2[1 - \lambda E(U_{r-1})]^2 + q_{r-1}/[1 - \lambda E(U_{r-1})]. \quad (20)$$

Since arrivals and departures take place one at a time, the number left in the r th queue by a departing customer must be distributed as Z . (This is a steady-state argument.) Let x be the number present in the r th queue at the moment when a customer of this queue is admitted to service, then

$$E(Z) = E(x) - 1 \quad \text{for } r \geq 2. \quad (21)$$

On the other hand, we have:

$$E(x) = (1 - \alpha_r) \sum_{i=r}^{\infty} E(K_i) \alpha_r^{i-r} + 1. \quad (22)$$

Substituting (12) and (15) in (22), summation and substitution in (21) yield

$$E(Z) = \alpha_r a / (1 - \alpha_r b) + [(1 - \alpha_r) / (1 - \alpha_r b)] E(K_r). \quad (23)$$

Substituting (20) and (23) in (19), we obtain

$$\begin{aligned} \sum_{i=1}^{\infty} E(T_i) = & [\lambda E(U_{r-1}^2) + \lambda E(Q_r^2)\beta_{r-1}/(1 - \alpha_r)]/2[1 - \lambda E(U_{r-1})]^2 \\ & + q_{r-1}/[1 - \lambda E(U_{r-1})] + [\alpha_r a / (1 - \alpha_r b)] \\ & + [(1 - \alpha_r) / (1 - \alpha_r b)] E(K_r) E(Q_r) / [1 - \lambda E(U_{r-1})]. \end{aligned} \quad (24)$$

Substituting (16) and (24) in (14), we have $E[T(l)]$ expressed as a function of $E(K_r)$ and of the various parameters of the process:

$$\begin{aligned} E[T(l)] = & [\lambda E(U_{r-1}^2) + \lambda E(Q_r^2)\beta_{r-1}/(1 - \alpha_r)]/2[1 - \lambda E(U_{r-1})]^2 \\ & + q_{r-1}/[1 - \lambda E(U_{r-1})] \\ & + [(b - \alpha_r) / \lambda \beta_{r-1}] [\alpha_r a + (1 - \alpha_r) E(K_r)] / (1 - \alpha_r b) \\ & + [E(K_r) - a / (1 - b)] (1 - b^{n-r}) / (1 - b) \end{aligned} \quad (25)$$

$$+(n-r)a/(1-b)\} + (n-r)a/(\lambda\beta_{r-1}) \\ + l - \sum_{i=1}^{n-1} \theta_i + \tau_r.$$

$E(K_r)$ is obtainable from

$$E(T) = \int_0^\infty E[T(l)]_{\mu} e^{-\mu l} dl, \quad (26)$$

where $E(T)$ is given by (7), $E[T(l)]$ for $n < r$ by (10), and for $n \geq r$ by (25). The result obtained for $E(K_r)$ is fairly complicated in form:

$$E(K_r) = [\lambda\beta_{r-1}(1-\alpha_r)(1-\alpha_r b)/(b-\alpha_r)] \\ \left\{ \sum_{n=r}^{\infty} \alpha_r^{n-r} \{ [\lambda E(U_n^2) + \lambda E(Q_r^2)\beta_{r-1}\alpha_r^{n+1-r}/(1-\alpha_r)] / \right. \\ \left. 2[1-\lambda E(U_{n-1})][1-\lambda E(U_n)] \right. \\ \left. + q_{n-1}/[1-\lambda E(U_{n-1})] \} - [1/(1-\alpha_r)] \right. \\ \left. \{ [\lambda E(U_{r-1}^2) + \lambda E(Q_r^2)\beta_{r-1}/(1-\alpha_r)] / 2[1-\lambda E(U_{r-1})]^2 \right. \\ \left. + q_{r-1}/[1-\lambda E(U_{r-1})] \} \right\} + a\alpha_r(1+b)/(b-\alpha_r). \quad (27)$$

APPENDIX FIFTEEN¹ Models with External Priority Proof of the Formulae

Priority queues

We now use the techniques of imbedded Markov² chains to analyze several classes of priority queues for the following stationary distributions: queue length, waiting time, queuing time, and busy period. These priority queues were previously defined and discussed within the context of service discipline. In order to specify these service disciplines, consider the previously discussed formulation of priority queues. There are N priority classes, each with independent Poisson processes of parameter λ_k , where $k = 1, 2, \dots, N$. These processes constitute inputs to a single-server system. Let the lower priority number indicate the higher priority, and let $H_k(x)$ be the service-time distribution for priority class k .

The modeling of a priority queueing system is a two-step procedure. We first treat the system as though no priorities are involved. Every caller is served on a first-come first-served basis. A queue-length generating function for an imbedded Markov chain previously discussed, is derived as shown in Appendix C for a queueing model having no priorities. We then take into consideration the effect of the priorities by modifying the results obtained in the first step to reflect the influence of caller priorities on the waiting-time distribution.

Preparatory to discussing the priority queueing systems, some additional concepts must be defined. Let the sum of the Poisson-process parameters be expressed as follows:

$$\Lambda_k = \sum_{i=1}^k \lambda_i$$

Each input is a Poisson process of parameter λ_k , therefore the sum Λ_k is also a Poisson process. Also, let the Laplace-Stieltjes transforms of the weighted service-time distributions be

$$\Psi_k(s) = \sum_{i=0}^k \frac{\lambda_i}{\Lambda_k} \psi_i(s) \quad (12)$$

and

$$\Phi_k(s) = \sum_{i=k+1}^N \frac{\lambda_i}{\Lambda_n - \Lambda_k} \psi_i(s)$$

Let a_i be defined as the first moment of the weighted service time, i.e.

$$a_k = \sum_{i=0}^k \frac{\lambda_i}{\Lambda_k} a_i$$

¹ Reproduced with the written permission of IBM Corporation. It is taken from:
W. Chan, "Single Server Queuing Processes in Computing systems," IBM Systems Journal, Vol. 9, No. 1, (1970), PP. 36-71

² The imbedded Markov Chains are described in this appendix.

In a priority queuing system, two priority-service disciplines are of interest—the preemptive-resume and the non-preemptive disciplines. Their stationary solutions are now presented.

In a preemptive-resume priority system, the presence of callers of priority numbers greater than k does not influence the stochastic law of the waiting time of callers with priority numbers less than or equal to k . Thus, callers with priority numbers greater than k are considered as not present in the system when studying the waiting time of callers with priorities less than or equal to k . To determine the stationary distribution of the waiting time for priority-class k , we first consider a modified queuing process with the following characteristics: a Poisson input of Λ_k , a service time distribution that is weighted by the input rates and characterized by $\psi_k(s)$ as defined in Equation 12, and a single server. The generating function can be obtained by using the technique of the imbedded Markov chain previously discussed. The Laplace-Stieltjes transform of the waiting time in this modified queuing process can be obtained

$$\Omega_k^*(s) = \frac{1 - \Lambda_k a_k}{1 - \frac{\Lambda_k [1 - \Psi_k(s)]}{s}} \quad (13)$$

If $\Lambda_k a_k$ is less than one, a stationary solution exists. If $\Lambda_k a_k$ is greater than or equal to one, there is no stationary solution³¹ for priority classes less than or equal to k .

The waiting-time distribution $W_k(x)$ for the priority- k caller is obtained from Equation 13 in the following way. During the waiting time of the modified queuing process, assume that there are j ($j = 0, 1, \dots$) arrivals of priority number less than k . The additional delay experienced by the caller with priority number k is then identical to the total of j independent busy periods in a single-server queuing process with Poisson input Λ_{k-1} and a service time distribution characterized by $\Psi_{k-1}(s)$. Let $D_{k-1}(x)$ be the distribution function of the length of a busy period in this process, and $\gamma_{k-1}(s)$ be the Laplace-Stieltjes transform of $D_{k-1}(x)$. From Equation E2 we can complete the formulation of the transform as follows:

$$\gamma_{k-1}(s) = \Psi_{k-1}\{s + \Lambda_{k-1}[1 - \gamma_{k-1}(s)]\} \quad (14)$$

Using similar reasoning to that given for Equation E1, we obtain from Equation 13 the following Laplace-Stieltjes transform of the waiting-time distribution for priority class k :

$$\Omega_k(s) = \Omega_k^*\{s + \Lambda_{k-1}[1 - \gamma_{k-1}(s)]\}$$

Using the same line of thought, the Laplace-Stieltjes transform $\theta_k(s)$ of the queuing-time distribution is found to be

$$\theta_k(s) = \Omega_k(s) \psi_k\{s + \Lambda_{k-1}[1 - \gamma_{k-1}(s)]\}$$

This equation shows that during the service of priority k callers under the preemptive-resume service discipline, other callers with priority numbers less than k may arrive and preempt the service.

nonpreemptive service

In the previously defined nonpreemptive service discipline, service time of a caller of any priority class is not interruptable. Consequently the presence of a low-priority call can affect the waiting time of a high-priority call. For example, if a low-priority call is being served when a call of a high-priority class arrives, the high-priority call must wait for completion of the lower-priority service before service begins. Calls of low priority receive immediate service when no calls of higher priority are waiting.

To find the waiting-time stationary distribution for priority class k , we use an approach similar to the one presented in the preemptive-resume priority queues. The generating function for this case, however, is slightly more complicated than the one for preemptive resume service. Consider a queuing process in which callers are classified into two queues. Let $\xi_n(k)$ be the queue length of calls having priority classes less than or equal to k , and let $\xi'_n(k)$ be the queue length of priority classes greater than k at the n th departure. The n th caller can be of any priority class. We now formulate the generating function for $\xi_n(k)$.

For a stationary process, $\xi_{n+1}(k)$ and $\xi_n(k)$ have the same probability distribution, and are related by

$$\xi_{n+1}(k) = \begin{cases} \xi_n(k) - 1 + \nu_{n+1} & \text{if } \xi_n(k) > 0 \\ \nu_{n+1} & \text{if } \xi_n(k) = 0 \text{ and } \xi'_n(k) = 0 \\ & \text{(i.e., both queues are empty,} \\ & \text{and the next call is of priority} \\ & \text{class less than or equal} \\ & \text{to } k) \\ \nu'_{n+1} & \text{if } \xi_n(k) = 0 \text{ and } \xi'_n(k) > 0 \\ & \text{(or the next call is of priority} \\ & \text{class greater than } k \text{ if } \xi_n(k) = \\ & \text{0 and } \xi'_n(k) = 0) \end{cases} \quad (15)$$

Here, ν_{n+1} is the number of new calls of priority classes less than or equal to k , if the $n + 1$ st service is of priority class less than or equal to k . The parameter ν'_{n+1} is the number of new calls of priority classes less than or equal to k if the $n + 1$ st service is of priority class greater than k .

Let $U_k(z)$ be the generating function of $\xi_n(k)$ so that

$$U_k(z) = \sum_{j=0}^{\infty} P\{\xi_n(k) = j\} z^j \quad (16)$$

Thus the probability that $\xi_n(k)$ is zero is expressed as follows:

$$P\{\xi_n(k) = 0\} = U_k(0)$$

Notice that Equation 15 expresses the following three mutually exclusive events:

- $\xi_n(k) > 0$, and the next arrival is of priority class less than or equal to k . The first event, $\xi_n(k) > 0$, is represented by the generating function

$$\frac{U_k(z) - U_k(0)}{z}$$

- $\xi_n(k) = 0$ and $\xi'_n(k) = 0$, and the next service is of priority class less than or equal to k . This event occurs with probability $(\Lambda_k/\Lambda_n)P_0$, where P_0 is the probability that the system is empty.
- $\xi_n(k) = 0$ and $\xi'_n(k) \geq 0$, and the next service is of priority class greater than k . The third event occurs with a probability of $U_k(0) - (\Lambda_k/\Lambda_n)P_0$, where P_0 is the probability that the system is empty, and

$$P_0 = 1 - \sum_{i=1}^n \lambda_i \alpha_i$$

Forming the generating functions on both sides of Equation 15 and using the technique of the imbedded Markov chain as given in the Appendices, we obtain Takacs' expression²⁹ for the generating function of $\xi_n(k)$

$$\begin{aligned} U_k(z) = & [U_k(z) - U_k(0)]\Psi_k[\Lambda_k(1 - z)] + \frac{\Lambda_k}{\Lambda_n} P_0 \Psi_k[\Lambda_k(1 - z)] \\ & + \left[U_k(0) - \frac{\Lambda_k}{\Lambda_n} P_0 \right] \Phi_k[\Lambda_k(1 - z)] \end{aligned} \quad (17)$$

Equation 17 generates the queue lengths of priority classes less than or equal to k at every departing instant, including the departure of those callers of priority classes greater than k . We now obtain a relation similar to the one in Equation D3 in Appendix D, which allows us to obtain the Laplace-Stieltjes transform of the waiting-time distribution from the queue-size generating function. We formulate the queue-size generating function observed by a departing caller of priority class less than or equal to k by Takacs' method.

In Equation 17, the $n + 1$ st customer is of priority class less than or equal to k , if the service-time distribution is of priority class less than or equal to k . Hence, the partial generating function of the right-hand side of Equation 17

$$\left[\frac{U_k(z) - U_k(0)}{z} + \left(\frac{\Lambda_k}{\Lambda_n} \right) P_0 \right] \Psi[\Lambda_k(1 - z)]$$

represents a departing caller of priority class less than or equal to k .

Takacs called this generating function $G^*(z)$. (Note that Takacs actually uses $G_k(z)$ in Reference 2; we add the asterisk to denote a modified queuing process.) The generating function of the queue size, $G_k(z)$, observed by a departure of priority class less than or equal to k , can be obtained from $G_k^*(z)$ by the following normalization:

$$G_k(z) = \frac{G_k^*(z)}{G_k^*(1)}$$

The Laplace-Stieltjes transform of the waiting time distribution of the modified queuing process is given by

$$G_k(z) = \Omega_k^*[\Lambda_k(1 - z)]\Psi_k[\Lambda_k(1 - z)]$$

If we replace $\Lambda_k(1 - z)$ by s , we obtain

$$\Omega_k^*(s) = \frac{1 - \sum_{i=1}^k \lambda_i \alpha_i + (\Lambda_N - \Lambda_k)\Phi_k(s)}{1 - \Lambda_k[1 - \Psi_k(s)]/s} \quad (18)$$

Finally, the Laplace-Stieltjes transform of the waiting time distribution for customers of priority class k is obtained in a way similar to the case of the preemptive-resume discipline as follows:

$$\Omega_k(s) = \Omega_k^*\{s + \Lambda_{k-1}[1 - \gamma_{k-1}(s)]\}$$

The Laplace-Stieltjes transform of the queuing time distribution is represented as

$$\theta_k(s) = \Omega_k(s)\Psi_k(s)$$

The transform takes this form because the service time of priority k cannot be interruptable during its service by any new arrivals of priority classes less than k .

waiting-time
moment
calculations

When a stationary solution exists (i.e., the Laplace transform of the waiting time exists), the moments such as the mean waiting time can be obtained by differentiating the Laplace transform as given in Equation 5. Let $a_k^{(r)}$ be the r th moment of the weighted service time distribution

$$a_k^{(r)} = (-1)^r \left. \frac{d^r}{ds^r} \Psi(s) \right|_{s=0} = \sum_{i=1}^k \frac{\lambda_i}{\Lambda_k} \alpha_i^{(r)}$$

where $\Psi_k(s)$ is given in Equation 12.

To find the waiting-time moments, the first step is to find the busy period moments by differentiating Equation 14. For example, the first two moments of the busy period distribution are the following:

$$d_{k-1}^{(1)} = \frac{a_{k-1}^{(1)}}{1 - \Lambda_{k-1}a_{k-1}^{(1)}}$$

$$d_{k-1}^{(2)} = \frac{a_{k-1}^{(2)}}{[1 - \Lambda_{k-1}a_{k-1}^{(1)}]^2}$$

The next step is to find an expression for the moment of the modified waiting time distribution, $W_k^{*(r)}$, by differentiating $\Omega_k^*(s)$ in Equation 18. Since we have two expressions for $\Omega_k^*(s)$, preemptive and non-preemptive, we have the following two cases:

Case 1 Preemptive-resume discipline

$$W_k^{*(1)} = \frac{\Lambda_k a_k^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]}$$

$$W_k^{*(2)} = \frac{\Lambda_k a_k^{(3)}}{3[1 - \Lambda_k a_k^{(1)}]} + \frac{\Lambda_k^2 [a_k^{(2)}]^2}{2[1 - \Lambda_k a_k^{(1)}]^2}$$

Case 2 Nonpreemptive discipline

$$W_k^{*(1)} = \frac{\Lambda_N a_N^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]}$$

$$W_k^{*(2)} = \frac{\Lambda_N a_N^{(3)}}{3[1 - \Lambda_k a_k^{(1)}]} + \frac{\Lambda_N a_N^{(2)} \Lambda_k a_k^{(2)}}{2[1 - \Lambda_k a_k^{(1)}]^2}$$

Finally, we compute the waiting-time moments from $\Omega_k(s)$ by differentiating as follows:

$$W_k^{(1)} = W_k^{*(1)}(1 + \Lambda_{k-1} a_{k-1}^{(1)}) = \frac{W_k^{*(1)}}{1 - \Lambda_{k-1} a_{k-1}^{(1)}}$$

and

$$W_k^{(2)} = W_k^{*(2)}[1 + \Lambda_{k-1} a_{k-1}^{(1)}]^2 + W_k^{*(1)} \Lambda_{k-1} a_{k-1}^{(2)}$$

$$= \frac{W_k^{*(2)}}{[1 - \Lambda_{k-1} a_{k-1}^{(1)}]^2} + \frac{W_k^{*(1)} \Lambda_{k-1} a_{k-1}^{(2)}}{[1 - \Lambda_{k-1} a_{k-1}^{(1)}]^3}$$

Since both $W_k^{*(1)}$ and $W_k^{*(2)}$ apply to Case 1 as well as Case 2, the following numerical examples are similarly separated into two cases.

Imbedded Markov chains

Since queuing is generally a stochastic process that is a function of the time parameter t , a class of stochastic processes useful for systems analysis is the *Markov process*. A queue is said to exemplify a Markov process if the present state of the system, including the queue, is sufficient to predict a future state (e.g., queue length) without knowledge of the past history of the system. By definition, Markov processes are continuous in time. When the system is studied at discrete time points, the collection of state probabilities constitutes a *Markov chain*. Here we are considering only qualitative properties of Markov chains. A mathematical definition of the Markov chain is given in Appendix A.

D. G. Kendall introduced the concept of an *imbedded Markov chain* because in practical cases queuing processes are not always Markovian nature. Kendall⁴ suggests that a non-Markovian process can be studied by extracting a set of points—called *regeneration points*—at which the Markov property exists. He formally defines a regeneration point as an epoch at which a knowledge of the state of the process has the characteristic Markovian consequence that a statement of the past history of the process loses its predictive value. A probabilistic definition of a regeneration point⁵ is as follows: an epoch is a regeneration point for the stochastic process $\{\xi(t)\}$ if and only if, for all $t > t_0$, $P\{\xi(t) \mid \xi(t_0)\} = P\{\xi(t) \mid \xi(\tau)\}$ for all $\tau \leq t_0$. A group of epochs, t_i (for $i = 0, 1, 2, \dots$), are regeneration points if and only if, for all $t_i < t < t_{i+1}$, $P\{\xi(t) \mid \xi(t_i)\} = P\{\xi(t) \mid \xi(\tau)\}$ for all $t_{i-1} < \tau \leq t_i$.

As an example, consider a queuing process having Poisson input (M) general service time (G), and a single server (1). (This is termed an M/G/1 queuing process by Kendall's queue-classification procedure.) It is Markovian if the present state of the process is described by the pair of random variables ξ and χ , where ξ is the instantaneous queue size, and χ is the expended service time of the customer who is currently being served. In general, however, the queue ceases to be Markovian if the state of the process is measured by the queue size alone. An exception occurs when the service time is exponentially distributed; a characteristic property of the exponential distribution ensures that a knowledge of the expended service time χ has no predictive value.

Illustrative of a set of regeneration points are the times at which callers depart from the system in the M/G/1 queuing process. Hence, the queue lengths at these departure times constitute a Markov

⁴ D. G. Kendall, "Stochastic Processes Occuring in the Theory of Queue and their Analysis by the Method of Imbedded Markov Chains," The annals of Mathematical Statistics, Vol. 4, (1952), .. 331-354.

chain with an enumerable infinity of states that can be studied through the theory of Markov chains. This technique, which is useful in many queuing systems including priority queues, is called the imbedded Markov chain technique because it involves the extracting of a discrete-time Markov chain imbedded in the continuous-time process.

Another important contribution of Kendall is a formal proof of the existence of a stationary solution for the M/G/1 queuing process if the *traffic intensity* (or system utilization) is less than unity. The traffic intensity ρ is defined as the product of the input rate λ and the mean service time α . Intuitively, when the traffic intensity is less than unity, the system should possess the characteristic "ergodic" property of settling down into an equilibrium mode of behavior independent of its initial state after the elapse of a sufficiently long period of time. When $\rho > 1$ no such behavior is to be expected. Kendall's proof uses Feller's general theory of recurrent events. The stationary solutions of priority queues are obtained by using the method of the imbedded Markov chain. They are basically extended solutions of Kendall's M/G/1 queuing process.

*** A P E N D I X S I X T E E N ***

PROGRAM FIM105 - FINDING THE ARRIVAL RATE FOR NEAR PERFECT
SERVICE

AUTHOR: PROSPER M. BERNARD
DATE: APRIL 29, 1972

```
// JOB FIM105
// OPTION CATAL,NODECK
// PHASE FIM105,*
// EXEC PL/I
* PROCESS STMT
/* AUTHOR PROSPER M BERNARD */
FIM105:
```

PROCEDURE OPTIONS (MAIN):

```
DCL L FIXED DEC (14,8) INIT (0);
DCL M FIXED DEC (14,8) INIT (0);
DCL MS FIXED DEC (14,8) INIT (0);
```

```
E = 5.0; /*LIMIT FOR WAITING */
E = 4.369; /* K, OF TOTAL EXPECTED SERVIU */
```

```
PUT EDIT (' ARRIVAL RATE FOR NEAR PERFECT SERVICE - ',
          ' EXPECTED WAITING TIME LESS THAN ',E)
          (PAGE,A,A,F(3,5));
```

```
PUT EDIT (' NUMBER OF DISKS ARRIVAL RATE MARGINAL VALUE')
          (SKIP (2),A);
```

```
PUT EDIT (' K L M ')
          (SKIP (1),A);
```

```
PUT EDIT (' ')
          (SKIP (1),A);
```

```
ET1 = 74.25; /* EXPECTED VALUE OF T1 IS ET1 ETC */
ER = 12.5; /* I.E. (25) + 0) / 2 */
FD = 0.64013;
ET2 = ER + ED;
VR = 25**2/12; /* VARIANCE OF R IS VR ETC */
VD = FD**2; /* EXPONENTIAL */
VT2 = VR + VD; /* ASSUMING INDEPENDENCE */
L = 100;
U1 = 1/ET1;
U2 = 1/ET2;
DE = 0.001; /* VALUE USED FOR DECREASE */
DE = 0.000001;
DE = 0.00000213;
MS = 0;
```

/* TIME IS IN MILLISECNDOS */

```

DO K= 1 TO 9:
/* START WITH L LARGE AND UNREALISTIC
AND WE TRY TO DECREASE IT
*/
L=1.0; /*1.E ACCESSSES PER MILLISECONDS */
L = 0.1;
L = 0.0060;
LOOP:
EW1 =(L/K)/(U1*(U1-(L/K)));
IF EW1 < 0 THEN GO TO DECR;
EW2 = (L / (U2**2))* (((U2**2)*VT2)+ 1) / ((2*(1 -(L/U2))));
IF EW2 < 0 THEN GO TO DECR;
EW = EW1 + EW2;
IF EW < E THEN GO TO OUT;
DECR:
L = L - DE;
GO TO LOOP;
OUT:
L = L * 1000 ;
M = L - MS;
MS= L ;
PUT EDIT
(K,L,4)
(SKIP (1),X(6),F(3,0),X(10),F(10,5),X(37),F(10,5));
END;
PUT EDIT (' TABLE 5.1.1')
(SKIP (3),A);
PUT EDIT ('SOURCE: OUTPUT OF THE PROGRAM FIM105 LISTED')
(SKIP (2),A);
PUT EDIT (' IN APPENDIX SIXTEEN')
(SKIP (1),A);
END;
/* END
// EXEC LINKEDT
/& END
// JOB FIM105
// EXEC FIM105
/& END

```

BIBLIOGRAPHY

- Wate, Joseph; Dubner, Harvey; and Weinberg, Sheldon. "Queuing Analysis of the IBM 2314 Disk Storage Facility," Journal of ACM, Volume 15, No. 4 (October 1968), pp. 577-590
- Askoff, R. L. (ed). Progress in Operations Research, Vol 1, New York: Wiley, 1961.
- Askoff, Russell, and Sasiem, Maurice. Fundamentals of Operations Research. New York: Wiley, 1968.
- Adiri, Igal. "A Time Sharing Queue with Preemptive Resume Priority Discipline." Operations Research, Statistics and Economic Mimeograph No. 7. Haifa, Israel: Israel Institute of Technology, May 1968.
- _____. "Computer Time-Sharing with Priorities." Operations Research, Statistics and Economic Mimeograph No. 13. Haifa, Israel: Israel Institute of Technology, May 1968.
- _____. "A Dynamic Time-Sharing Priority Queue." Journal Of ACM, Volume 18, No. 4, (October 1971), pp. 603-610
- _____. "A Note on Some Mathematical Models of Tame-Sharing System." Journal of ACM, Volume 18, No. 4, (October 1971), pp. 611-615.
- Adiri, I., and Avi-Itzhak, B. "A Time-Sharing Queue with a Finite Number of Customers," Journal of ACM, Vol. 16, No. 2, (April 1968), pp. 315-324.
- _____. AND _____. "A Time-Sharing Queue with a Finite Number of Customers." Operations Research, Statistics and Economic Mimeograph no. 10. Haifa, Israel: Israel Institute of Technology, February 1968.
- _____. AND _____. "A Time-Sharing model with many Queues." Operations Research, Vol. 17, No. 6, (November-December 1969), pp. 1077-1089.
- Allan, H., and Anderson, J. An Electronic Cash Credit System. New York: American Management Association, 1965.
- Aron, J. D. "Real-Time System in Perspective." IBM System Journal, No. 1 (1967), pp. 49-67.
- Aronofsky, J. S. Progress in Operations Research, Vol. 111. New York: John Wiley & Sons, Inc., 1969, (Chapter 2 & 3)

- Balachandran, K. R. "Queue Length Dependent Priority Queues." Management Science, Vol. 17. No. 7, (March 1971) pp. 463-37.
- Balintfy, J. L. "Menu Planning By Computer" Communication of ACM, Vol. 7, No. 4, April 1964. pp. 255-259.
- Barret, Cecil C. The Future of Computer Utility. New York: American Management Association, 1967.
- Brockmeyer, E., Halstron, H. L., and Jensen, A. "The Life and Works of A. K. Erlang," Transactions of the Danish Academy of Science (Copenhagen: Akademiet for de Tekniske Videnskaber), Vol. 2, (1948)
- Brosh, I. "The policy Space Structure of Markovian System with two Types of Service," Management Science, May 1970, pp. 607-622.
- Brown, Theodore. A Network Queuing Model of Multiprogramming Time-Shared Computer System. PhD dissertation, New York, University, 1971.
- Burck, Gilbert. The Computer Age. New York: Harper & Row, 1965.
- Burdick, D., Sasser, W., and Naylor, T. "The Application of Sequential Sampling to Simulation: an Example Inventory Model." Communication of the ACM, Vol. 13, No. 5. (May 1970), pp. 289-296
- Burke, P. J. "The Output of a Queuing System." Operations Research. Vol. 4, (December 1956), pp. 699-704.
- Certaine, J. "On Sequences of Pseudo-Random Numbers of Maximal Length", Journal of ACM, Vol. 5, 1968, pp. 353-356.
- Chang, W. "Queue With Feedback for Time-Sharing Computer System Analysis." Operations Research 16, (1968), pp. 613-627.
- _____. "A Queuing Model for a Simple Case of Time-Sharing." IBM Systems Journal, Vol. 5, No. 2, (1966) pp. 115-126.
- _____. "Single Server Queuing Processes in Computing Systems." IBM Systems Journal, Vol. 9, No. 1, (1970) pp. 36-71
- _____. Computer Congestion Analysis: A Theoretical Treatment. PhD dissertation, Polytechnic Institute of Brooklyn, 1969.
- Clark, S. R. and Rourke, T. A. "A Simulation Study of Cost of Delay in Computer Systems." Proceeding of the Fourth Conference on Applications of Simulation, New York, 1970.

- Clarke, C. E. "Importance of Sampling in Monte Carlo Analyses." Operations Research, Sept. 1961.
- Clarkson, G. P. and Melzer, A. H. "Portfolio Selection: A Heuristic Approach." The Journal of Finance, Vol. 15 No. 4, Dec. 1960, pp. 465-480.
- Coffman E. G. "Studying Multiprogramming Systems with the Queuing Theory," Datamation, Vol. 13, No. 6 (June 1967). pp. 47-54
- _____. "Analysis of Drum Input/Output Queue Under Scheduled Operation in a Paged Computer System," Journal of ACM, Vol. 16, No. 1, (January 1969), pp. 73-91.
- _____. "An Analysis of Computer Operations Under Running-Time Priority Disciplines." Proc. ACM Symp. on Interactive Systems for Experimental and Applied Mathematics, Academic Press, New York, 1968.
- _____. "Analysis of Two Time-Sharing Algorithms Designed for Limited Swapping." Journal of ACM, Vol. 15, No. 3, (July 1966) PP. #3L_353.
- _____. "Stochastic Models of Multiple and Time-Shared Computer Operations. No. 66-38, UCLA Eng. Rep., UCLA, Los Angeles, Calif., June 1966.
- Coffman, E. and Kleinrock, Leonard. "Feedback Queuing Models for Time-Shared Systems," Journal of ACM, Vol. 15, No. 4 (October 1968), pp. 549-576.
- _____. AND _____. "Computer Scheduling Methods and Their Countermeasures." Proc. AFIPS 1968 Spring Joint Comput. Conf., Vol. 32, Thompson Book Co., Washington, D. C., pp. 11-21
- Coffman E. G., and Krishnamoorthi, B. "Preliminary Analyses of Time-Shared Computer Operation." Doc. SP-1719, System Development Corp., Santa Monica, Calif., August 1964
- Coffman, E. G. and Muntz, R. A. "Preemptive Scheduling of Real-Time Tasks Multiprocessor Systems." Journal of ACM, Vol. 17, No. 2, pp. 324-338.
- Cohen, J. W. "On the Queuing Processes of Lines." Philips Techn. Dept. 1956.
- Corbato, F. J., "An Experimental Time Sharing System." Proc. AFIPS 1962 Spring Joint Compt. Conf., Vol. 21, Spartan Books, New York, pp. 335-344
- Chandy, K. M. and Ramamoorthy, C. V. "Optimization of Memory Hierarchies in Multiprogrammed Systems." Journal of ACM Vol. 17, No. 3, (July 1970), pp. 426-445.

- Desmonde, William H. Real-Time Data Processing Systems. Englewood Cliffs, New Jersey: Prentice Hall, 1964.
- Denning, P. J. "The Working Set Model for Program Behavior." Comm. ACM, Vol. 11, No. 5 (May 1968), pp. 323-333.
- _____. "A Statistical Model for Console Behavior in Multi-user Computers," Communications of the ACM, Vol. 11, No. 9, (September 1968), pp. 605-612.
- _____. "Resource Allocation in Multiprocess Computer Systems." Rep. MAC-TR-50, MIT Project MAC, Cambridge, Mass., May 1968.
- Delbrouk, L. E. "A Feedback Queuing System with Batch Arrivals, Bulk Service, and Queue Dependent Service Time." Journal of the ACM, Vol. 17, No. 2, (April 1970) pp. 314-323.
- Dexter, Albert, Evaluation of Time-Shared Systems: An Approach Towards Processing Optimization. PhD Dissertation, Columbia University 1970.
- Di Roccaferri, Ferro. Operations Research Models. New York: South Western, 1964.
- Elliot, John's. "The On-Line Real Time Environment." Technical Information Exchange, No. 4-0147, White Plains, New York, 19
- Emshoff, E., & Sisson, R., Computer Simulation Models, New York: MacMillan, 1970.
- Estrin, G., and Kleinrock, L. "Measures, Models, and Measurement for Time Shared Computer Utilities." Proc. ACM 1967 Nat. Conf., Thompson Book Co., Washington, D.C. pp. 85-96
- Feller, W. An Introduction to Probability and Its Applications. Vol. 1, New York: John Wiley & Sons, 1957.
- _____. An Introduction to Probability Theory and Its Application Vol. 11, New York: John Wiley & Son, 1971.
- Fife, D. W., and Smith, J. L. "Transmission Capacity of Disk Storage Systems with Concurrent Arm Positioning." IEEE Trans. EC, 14, No. 4 (August 1965), pp. 575-582.
- Fishman, George "Estimating Sample Size in Computing Simulation Experiments." Management Science, Vol. 18, No. 1, (Sept. 7) pp. 21-35
- Fishman, G. and Kiviat, Philip. "The Analysis of Simulation-Generated Time series." Management Science, Vol. 13, No. 7, (March 1967), pp. 525-557.
- Forgarty, D. W. "Simulation: A Decision Making Technique." Production and Inventory Management, Oct. 1967, pp. 69-82
- Gonzalez, Richard, and McMillan, Richard. Systems Analysis: A Computer Approach to Decisions Models. Homewood Illinois: Richard Dornan Inc., 1968.

024

Gordon, G. "A General Purpose System Simulator." IBM Systems Journal, (September 1962).

_____. System Simulation. Englewood Cliff, New Jersey: Prentice Hall, 1969.

_____. "Simulation Languages for Discrete Systems." Proceedings IBM Scientific Computing Symposium Simulation Models and Gaming, White Plains, N.Y.: IBM, 1966 pp. 101-118

_____. "A General Purpose Simulator." IBM Systems Journal, Sept. 1962.

_____. System Simulation, Englewood Cliff, N.J.: Prentice-Hall, 1969.

Graham W. et al. "The Application of Sequential Sampling to Simulation: an Example inventory Model." Communication OF THE ACM, Vol. 13, No. 5, (May 1970), pp. 289-296

Greenberger, M. "the Priority Problem and Computer Time Sharing." Man. Sci. 12 (1966), pp. 888-906.

Gruenberger, Fred, The Transition to On-Line Computing, Washington: Thompson Book, 1967.

Hammersley, J. M. and Handcomb, D. C. Monte Carlo Methods, New York: John Wiley, 1964.

Hanssmann, W, Kistler, W, and Schultz, H. "Model for Computer Center Planning." IBM Systems Journal, Vol. 10, No. 4, (December 1971) pp. 305-324.

Heacox, Harry and Purdom, Paul. "Analysis of Two Time-Sharing Queuing Models," Journal of The ACM, Vol. 19, No. 1, January 1972, pp. 70-91.

Head, Robert. Real-Time Business Systems. New York: Holt Rinehart and Winston, 1964.

Hellerman, H. "Some Principles of Time-Sharing Scheduler Strategies." IBM Systems Journal, Vol. 8, No. 2, (1969) pp. 94-118.

Hillier, Frederick S. "The Derivation of Probabilistic Information for Evaluation of Risky Investments." Management Science, Vol. 9, April 1963. pp. 443-457

Hillier, F., & Lieberman, G. Introduction to Operations Research, San Francisco: Holden-Day, Inc., 1969 (Chap 14).

International Business Machines. IBM/360 - 2314 Direct Access Storage Facility, New York: IBM, Ref. A26-3599, 1955.

_____. Random Number Generation and Testing. IBM, Ref. C 20-8011, 1959.

_____. General Purpose Simulator. New York: IBM, ref. H 20-6326, 1967.

- _____. General Purpose Simulator. New York: IBM, ref. B 20-634
1967.
- _____. IBM/360 System Description. New York: IBM, ref.
A 26-5988, 1969
- _____. IBM/360 Direct Access Devices. New York: IBM, ref.
C 20-1649, 1968.
- IBM. IBM/360 Model 30 Channel Characteristics and Functional
Evaluation. New York: IBM, ref. A 24-3411, 1969
- _____. QTAM Message Control Program. New York: IBM,
Ref. C 20-5004, 1969.
- _____. IBM Teleprocessing Summary, New York: IBM, Ref.
A 24-3080, 1968
- _____. Data Communication Concepts and Communication
Facilities. New York: IBM, Ref. E 20-8158, 1968.
- _____. GPSS/360 Operators Manual. Poughkeepsie, New York:
IBM, Ref. H 20-0327, 1968
- _____. GPSS/360 System Manual. Poughkeepsie, New York:
IBM, Ref. Y 20-0075, 1968.
- _____. Analysis of Some Queuing Models in Real Time Systems.
New York: IBM, Ref. F 20-0007, 1966
- _____. IBM 360 Model 30 Functional Characteristics. New York:
IBM, Ref. A 24-3231, 1969
- _____. Capital Investment Studies Using GPSS. IBM Ref.
GE 20-0313-1
- _____. Proceedings IBM Scientific Computing Symposium Simulation
Models and Gaming. White Plains, New York. IBM 320-1940-0
1966.
- _____. IBM/360 System, New York: IBM, Ref. S 20-0903, 1968
- _____. A Data Processing Glossary. New York: IBM, Ref.
C 20-1699, 1969
- _____. GPSS Education Guide. Poughkeepsie, New York: IBM
R 20-4075, 1968
- _____. An Introduction to Simulation by Computer, Poughkeepsie,
New York: IBM, ref. E 20-0272, 1968.

IBM. GPSS/360 Introduction User's Manual, Poughkeepsie,
New York: IBM, Ref. H20-0304, 1968

_____. GPSS/360 User's Manual. Poughkeepsie, New York:
IBM. Ref. H20-0326, 1968.

Irani, K. Wallace, V. "On Network Linguistics and The
Conversational Design of Queuing Network." Journal of ACM
Vol. 18, No. 4, (October 1971), pp. 616-629

Jackson, J. R. "Jobshop-like Queuing Systems." Man. Sci. IQ
(1963), pp. 131-142

Kabak, Irwin. "Blocking and Delays in M/M/C Bulk Arrival
Queuing Systems." Management Science, Vol. 7, No. 1,
(September 1970), pp. 112-125

Kendall, W. G. "Some Problems in the Theory of Queues." Journal
of the Royal Statistical Society, Series B, Vol. XIII, No. 2

Kistler, W., Hanssmann, W. and Schultz, H. "Model for Computer
Center Planning." IBM Systems Journal, Vol. 10, No. 4,
(December 1971), pp. 305-324.

Kiviat, P. and Fishman G. "The Analysis of Simulation Generated
Time-Series." Management Science, Vol. 13, No. 7,
(March 1967), pp. 525-557

Kleinrock, L. "A Conservation Law for a Wide Class of Queuing
Disciplines." Nav. Res. Log. Quart. 12 (1965) pp. 181-192

_____. "Time Shared Systems: A Theoretical Treatment."
Journal of ACM, Vol. 14, No. 2 (April 1967) pp. 242-261

_____. "Certain Analytic Results for Time Shared Processors."
Proc. 1968 IFIP Cong., 1969

_____. "Communication Nets: Stochastic Message Flow and Delay."
McFraw-Hill, New York, 1964

_____. "Sequential Processing Machines (S.P.M.) Analyses with a
Queuing Theory Model." Journal of ACM, Vol. 13, No. 2
(April 1966), pp. 179-193

_____. "Optimum Bribing for Queue Position." Operational Research,
Vol. 15 (1967) pp. 304-318

_____. and Coffman, E. G. "Distribution of Attained Service in
Time Shared Systems." Journal of Computer System Science,
Vol. 1 (1967), pp. 287-298

_____. "Analysis of a Time Shared Processor." Nav. Res. Log. Quart.
Vol. 11 (1964) pp. 59-73

Krishnamoorthi, B. "The Stationaty Behavior of a Time Sharing
System under Poisson Assumption." Doc. SP. 2090, System
Development Corp., Santa Monica, California, Sept. 1965.

_____. and Wood, R. C. "Time Shared Computer Operations with both
Inter-Arrival and Service Times Exponential." Journal of ACM
Vol. 13, No. 3 (July 1966), pp. 317-338

- Krolak, Patrick. "Market Simulation: Investment Games Computer Play." Computer Decisions. November 1969. pp. 25-30
- Lee, Alec. Applied Queuing Theory. Montréal, Québec: MacMillan, 1966.
- LePage, W. R. Complex Variable and Laplace Transform for Engineers. New York: McGraw-Hill, 1961.
- Little, J. D. C. "A Proof of the Queuing Formula $L = \lambda W$." Operations Research, Vol. 9 (1961), pp. 383-387.
- Lowe, Thomas, C. "The Influence of Data Base Characteristics and Usage on Direct Access file Organization." Journal Of ACM, Vol. 15, No. 4 (October 1968), pp. 514-534
- MacDougall M. H. "Computer System Simulation: an Introduction." Computing Survey, Vol, 2, No. 3 , Sept 1970, pp. 191-208
- Malcolm, D. G. Management Control System. New York: John Wiley, 1961
- Margenthaler, George W. Progress in Operations Research. New York: John Wiley, 1961
- Markowitz, H., Haaumer, B.; and Karr, H. Simscrip: A Simulation Programming Language. Washington: Prentice Hall, 1968
- Martin, F. Computer Modeling and Simulation, New York: McGraw Hill, 1969
- Martin, James. Programming Real Time Systems. Englewood Cliffs, N.J.: Prentice Hall Inc., 1965
- _____. Design a Real Time Computer System. Englewood Cliff, New Jersey: Prentice Hall Inc., 1967
- _____. Telecommunication and the Computer. Englewood Cliff New Jersey: Prentice Hall, 1969
- McAulay, S. E. "Job Stream Simulation using a Channel Multiprogramming feature." Proceedings of the Fourth Conference on Application of Simulation, New York, 1970 pp. 190-194
- McCredie, John W., and Schlesinger Steven J. "A Modular Simulation of TSS/360" Proceedings of the Fourth Conference on Applications of Simulation; New York, 1970, pp. 200-205
- McKinsey & Company. Unlocking the Computer's Profit Potential. New York: McKinsey & Company, 1968
- McMillan, Claude, and Gonzalez, Richard. Systems Analysis: a Computer Approach to Decisions Models. Homewood Illinois: Richard Irwin Inc., 1968.

- McLead, John. Simulation: Modeling of Ideas and Systems with Computers. New York: McGraw-Hill, 1968.
- McKinney, J. M. "A Survey of Analytical Time-Sharing Models." Computing Surveys, Vol. 1, No. 2 (June 1969)
- Mehra, Basant K. "Computer Simulation versus Analytical Techniques as Applied to Mine Haulage Systems." Bulletin of the Information Processing Society of Canada, Vol. 9, No. 1, (winter 1969), p21-
- Meyers, Charles, The Impact of Computers on Management. Cambridge, Massachusetts: MIT Press, 1966.
- Moy, William. Sample Techniques for Increasing the Efficiency of Simulation Queuing Systems. PhD Dissertation, Northwestern University, 1965.
- Muntz R. R., and Coffman E. G. "Preemptive Scheduling of Real-Time Tasks Multiprocessor Systems." Journal of ACM, Vol. 7, No. 2, pp. 324-338
- Newell G. F. and Haji Rasoul. "Optimal Strategies for Priority Queues with Non-Linear Costs of Delay." SIAM Journal of Applied Mathematics. March 1971, p. 224, Vol. 20, No. 7
- Miller L. W., and Schrage, L. E. "The Queue M/G/1 with the Shortest Remaining Processing Time Discipline." Operations Research, Vol. 14, No. 4 (July-August 1966) pp. 670-684
- Milch, P. R., and Wagganer M. H. "A Random Walk Approach to a Shutdown Queuing System." SIAM Journal on Applied Mathematics. July 1970
- Naylor, Thomas H. Computer Simulation Experiments with Models OF Economic System. New York: Wiley & Sons, 1971
- _____. and Finger J. M. "Verification of Computer Simulation Models." Management Science, (October 1967) pp. B92-B101
- Naylor, T., Wertz K., and Wonnacott T. "Methods for Analysing Data from Computer Simulation Experiments." Communication of ACM, Vol. 10, No. 11, (November 1967), pp. 703-710
- Naylor, T. et al. "The Application of Sequential Sampling to Simulation: an Example Inventory Model." Communication of ACM, Vol. 13, No. 5. (May 1970) pp. 287-296
- Nelbrouk L. N. "A Feedback Queuing System with Batch Arrivals Bulks Service, Queue Dependent, Service Time." Journal of ACM, April 1970, p. 314.
- Nielsen, Norman R. "An Analysis of Some Time-Sharing Techniques." Communication of the ACM, Vol. 14, No.2, February 71. pp. 79-90
- North Atlantic Treaty Organization. Queuing Theory. New York: American Elsevier Publishing Company, Inc., 1965

Ockene, Arnold. "Can Computer Simulation Work for You." Decisions, January 1970, pp. 34-37

Orcutt, G. H. "Simulation of Economics Systems." The American Economic Review, Vol. 1, No. 5, December 1960, pp. 893-907

Panico, Joseph A. Queuing Theory. Englewood Cliffs: Prentice-Hall, 1969

Patel N. R. "A Mathematical Analysis of Computer Time-Sharing Systems." Int. Techn. Rep. No. 20 ARO (D), Grant no. Da-ARO (D)-31-124-G158, Oper. Res. Center, MIT, Cambridge, Mass, 1961

Phipps, T. E. "Machine Repair as a Priority Line Problem," Operations Research, Vol. 4, 1956, pp. 76-86

Purdom, Paul and Heacox, Harry. "Analysis of two Time-Sharing Queuing Models," Journal of ACM, Vol. 19, No. 1, January 1972, pp. 70-91

Prabhu, N. U. Queues and Inventories, New York: Wiley, 1965

Ramanoorthy L. V., and Chandy K. M. "Optimization of Memory Hierarchied in Multiprogrammed Systems." Journal of ACM Vol. 17, No. 3, July 1970, pp. 426-445.

Rasch, Philip J. "A Queuing Theory Study of the Round Robin Scheduling of Time Shared Computer Systems." Journal of ACM Vol. 17, No. 1, January 1970, pp. 131-145

_____. A Queuing Theory Study of Time-Shared Computer Systems. PhD dissertation, Southern Methodist University, 1967.

Raychaudhuri, D. K. "Combinatorial Information Retrieval System." SIAM Journal of Applied Mathematics, Vol. 16, No. 5 (September 1968) pp. 973-992

Reich E. "Waiting Time when Queues are in Tandem." Ann. Math. Statist. Vol. 28, No. 3, (september 1957), p. 768

Rockwell T. H., Barnum R. A., and Griffin W. C. "Inventory Analysis as Applied to Hospital Whole Blood Supply and Demand." The Journal of Industrial Engineering, Vol. 13, No. 2, March and April 1962, pp. 109-114

Rothstein M. F. Design of Real-Time Systems. New York: John Wiley & Sons, Inc., New York 1970

Rourke, T. A. and Clark S. R. "A Simulation Study of Cost of Delays in Computer Systems." Proc. of the Fourth Conf. on Application of Simulation, New York 1970

Saaty T. L. "Resume of Useful Formulas in Queuing Theory." Operations Research, Vol. 5, No. 2 (1957)

_____. Elements of Queuing Theory. New York: McGraw Hill, 1961

Sasieni Maurice and Yaspan A. Operations Research N.Y.: Wiley, 1961

- Sasieni M., Ackoff R. Fundamentals of Operations Research. New York: Nohn Wiley & Sons, 1968
- Sasser W. E., Burdick, D., Graham W., and Naylor T. "The Application of Sequential Sampling to Simulation: An Example Inventory Model." Communications of the ACM, Vol. 13, No. 5 (May 1970) pp. 287-296
- Schatzoff M. Margolin B., and Parmelee R. "Analysis of Free Storage Algorithms." IBM Systems Journal. Vol. 10, No. 4 (December 1971), pp. 283-305
- Shedler C. S., and Yang S. C. "Simulation Model of Paging System Performance." IBM Systems Journal, Vol. 10, No. 2 (July 1971) pp. 113-128
- Schelsingers J. and McCredie J. "A Modular Simulation of TSS/360. Proc. of the Fourth Conf. on Applications Simulation. New York 1970, pp. 200-205
- Shemer, J. E. "Some Mathematical Considerations of Time-Sharing Scheduling Algorithms." Journal of ACM, Vol. 14, No. 2 April 1967, pp. 262-272
- Scherr A. L. An Analysis of Time Shared Computer Systems, Cambridge, Massachusetts: MIT Press, 1967.
- Schmookler M. Queuing Models for Computer Storage Problems. PhD Dissertation, Princeton University, 1969
- Schrage L. E. and Miller L. W. "The Queue M/G/1 with the Shortest remaining Processing Time Discipline." Operations Research, Vol. 14, No. 4 (July-August 66), pp. 670-684
- Schrage, L. "Analysis and Optimization of a Queuing Model of a Real-Time Computer Control System." IEEE Computer Group Conference Digest. June 1969. pp. 34-36
- Schultz H., Hanssmann W., and Kistler W. "Model for Computer Center Planning." IBM Systems Journal, Vol. 10, No. 4 December 1971, pp. 303-324
- Scott McKinley. "Queuing with Control on the Arrival of Certain Type of Customers." CORS Journal, Vol. 8, No. 2 July 1970, pp. 75-86
- Seaman P. H., and Soucy R. C. "Simulating Operating Systems." IBM Systems Journal, Vol. 8, No. 4 1969 pp. 264-279
- Smith W. L. "On the Distribution of Queuing Times." Proc. Cambridge Phil. Soc., 1953, pp. 449-461
- Sippl Charles. Computer Dictionary and Handbook. New York: Howard W. Sams & Co. Inc., 1966
- Shreider Y. The Monte Carlo Method. New York, Pergamon Press, 1

- Stimler, Paul. "Some Criteria for Time-Sharing." Communications of The ACM, Vol. 12, No. 1 (January 1969), pp. 47-54
- _____. Real-Time Data Processing Systems. New York: McGraw Hill, 1969.
- Strauch Ralph. "When Queues look the Same to an Arriving Customer as to an Observer." Management Science, Vol. 17, No. 3, November 1970. pp. 140-141
- Tacher K. D. "A Review of Simulation Languages." Operations Research Quarterly. Vol. 16, No. 2, (June 1965)
- Takacs, L. Introduction to the Theory of Queues. Oxford U. Press, New York, 1962.
- _____. "A Single Server Queue with Feedback." Bell System Technical Journal, Vol. 42, No. 2, March 63, pp. 503-509
- _____. "Delay Distributions for one line with Poisson Input General Holding Times, and Various Orders of Service." Bell System Technical Journal, Vol. 42, No. 2, March 1963
- _____. "Investigation of Waiting Lines." ACTA Math. and Sci., 195
- _____. Combinatorial Methods in the Theory of Stochastic. New York: John Wiley, 1967.
- Teichroew D., and Lubin J. F. "Computer Simulation Discussion of the Techniques and Comparison of Languages." Communications the ACM, Vol. 9, No. 10, Oct. 1966.
- Thorne J. F. Real-Time Systems in Accounting Applications, Data Management, Vol. 8, No. 1, January 1970, pp. 28-33
- Uttal William R. Real-Time Computers. New York: Harper & Row, 1
- Wallace, V. L. and Mason, D. L. "Degree of Multiprogramming in Page on Demand Systems." Communications of the ACM, Vol. 12, No. 6, June 1969, pp. 303-309
- Wallace V. and Irani K. "On the Network Linguistics and the Conversational Design of Queuing Networks." Journal of ACM, Vol. 18, No. 4, (October 1971), pp. 616-629.
- Williams, John, G. "Storage Utilization in a Memory Hierarchy when Storage Assignment is Performed by a Hashing Algorithm." Communications of the ACM, Vol. 14, No. 3, March 1971, pp. 172-175
- Weild, Martin H. Dictionary of Computer Terms. New York: Hayden, 1969.

Wertz K., Naylor T., and Wonnacott T. "Methods for Analysing Data from Computer Simulation Experiments." Communications of the ACM, Vol. 10, No. 11, (November 1967), pp. 703-710

Winer B. J. Statistical Principles in Experimental Design. New York: McGraw Hill, 1962.

_____. Statistical Principles in Experimental Design. New York: McGraw Hill, 1971

Wonnacott T., Wertz K. and Naylor T. "Methods for Analysing Data from Computer Simulation Experiments." Communications of the ACM, Vol. 10, No. 11, (November 1967), pp. 703-710

Yaspan, Arthur, and Sasieni, Maurice. Operations Research. New York: John Wiley & Sons, 1959.

Ziecler, James. Time Sharing Data Processing Systems. Englewood Cliffs, New Jersey: Prentice Hall, 1967.

Yang S. C., and Sholder G. S. "Simulation Model of Paging System Performances." IBM System Journal, Vol. 1, No. 2, July 1971, pp. 113-128.

Vanhorne R. "Validations of Simulation Results." Management Science, Vol. 17, No. 15, January 1971, pp. 247-258

AUTOBIOGRAPHY

Prosper M. Bernard, the youngest of thirteen children, was born in St. Jean, Québec on August 15, 1941. He received his primary and secondary education in Granby, Québec where the family lived.

He received a B.A. degree summa cum laude from the University of Montreal (at College St. Marie) and a B.Sc. degree from McGill University. He attended the evening division of the graduate school of McGill University.

He was admitted with his wife into the United States in 1968 and since then, he has received an MBA from St. John's University and a CDP (Certificate in Data Processing) from the DPMA. The degree of Doctor of Philosophy is expected.

Mr. Bernard held teaching positions both in Québec and in New York. He also has a long experience in the information processing field as a programmer, systems analyst, management consultant, and manager. He is a member of many professional societies.

Mr. Bernard and his wife, the former Francine N. Gascon, have two sons, Erik and Prosper jr.