AN ABSTRACT OF THE THESIS OF

MARY NAGUIB YOUSSEF for the DOCTOR OF PHILOSOPHY
(Name) (Degree)

in STATISTICS presented on March 2, 1971
(Major) (Date)

Title: OPTIMIZATION TECHNIQUES FOR TIME-SHARED COMPUTER SYSTEMS

Redacted for Privacy

Abstract approved:
Donald Guthrie, Jr.

The inefficiency of time-shared computer systems compared to batch processing systems is in the time lost in swapping operations. The larger the allocated quantum size, the less swap time is incurred. In order to guard against intolerable response time while lengthening the quantum size, the response time of a common request must be regulated. The criteria used in this paper to regulate the response time is to vary the quantum, with the number of users in the system, in such a way that the computer response time approaches the human response time. Based upon this concept, models are designed and analyzed to design an optimal scheduling algorithm which allocates the quantum dynamically.

The models proposed are based upon Markovian assumptions for both arrival and service times. The priority discipline is round robin with dynamic quantum allocation. The swap time is assumed

to be constant and the overhead time is zero. The inverse measure of performance is assumed to be the expected square difference between the cycle time and the mean human response time.

In order to optimize these models two techniques are discussed. In the first, a mathematical optimization model is formulated in which a Markov chain is imbedded at the epochs of the beginning of a cycle. The cost function is assumed to be the inverse measure of performance. A technique suggested by Howard for optimizing a stochastic system under Markovian assumptions provides an optimal policy by which the scheduling algorithm allocates the quantum. The second technique discussed is based upon an optimal control system approach. The quantum size is chosen in such a way as to assure some stability property while improving system performance. A numerical example which illustrates these methods is provided.

Optimization Techniques for Time-Shared Computer Systems

by

Mary Naguib Youssef

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

June 1970

APPROVED:

Redacted for Privacy

Associate Professor of Mathematics and Statistics

in charge of major

Redacted for Privacy

Chairman of Department of Statistics

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 12, 1970

Typed by Clover Redfern for Mary Naguib Youssef

ACKNOWLEDGMENT

I would like to express my gratitude to my major professor, Dr. Donald Guthrie, Jr. for his advice and continual support during my graduate program. I am deeply indebted to Dr. V. J. Bowman for his valuable assistance and advice in the research for this thesis and in the preparation of the manuscript. My thanks go to Dr. H. D. Brunk and Dr. L. C. Hunter for their helpful comments, to the Department of Statistics at Oregon State University and the National Science Foundation for the financial support.

I would like also, to express my sincere appreciation to my daughter, Amany, for her patience and understanding over the years of my graduate work, and to dedicate this work to the memory of my father, Naguib Youssef, who was my mentor throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

OPTIMIZATION TECHNIQUES FOR TIME-SHARED COMPUTER SYSTEMS

I. INTRODUCTION

Although a great deal has been published on the analysis of probability models of time sharing computer operating systems, the problem of providing an optimization model is far from being solved. Time sharing systems are in general more sophisticated than one can analyze with simple queuing theory. Mathematical models of such systems usually lack sufficient detail and have limited applicability, yet many of the characteristics of time sharing systems have been summarized by those models.

In order to facilitate later discussions, a brief description of a simple time sharing environment, discussions of the critical problems of the system which are facing the designer in his experimenting and the different approaches used to solve these problems follow.

The Time Sharing System

A data processing system may be viewed as a set of terminals, memories and processors all interconnected by a network of communication channels. A time sharing system is a data processing system which is characterized by its ability to service multiple users simultaneously. A user is able to interact with the computer through

a terminal and at unscheduled times. An interaction consists of the user requesting and then receiving service from the system. A user's request may be viewed as a program needing to be served by the central processor and other devices. The events usually forming an interaction are, the user thinking, supplying new input from his terminal, waiting for response from the system, and finally getting output. The response time of the system is defined as the time between receipt of a specified request by the system and the satisfaction of that request at the terminal. The human response time is the interval between the response of the system to a user's request and the insertion of his next request. With these definitions, one may think of the system as being in one of two states: either the user is waiting for the system to respond or the system is waiting for the user to respond.

In general, the storage space of a time sharing system can be shared by more than one program at any time. However, the processor can serve only one program at any instant. Thus in order to ensure the desired concurrency in servicing the requests generated by a population of users at the same time, these requests will enter a queue upon their arrivals, and a member of the queue chosen by a defined scheduling algorithm will be allowed to use the processor for a maximum period of time called a quantum. If one quantum is sufficient to process the entire request, the request then leaves the

system. If not, service is interrupted, the request reenters the queue partially processed, and waits until the scheduling algorithm decides to give it another turn of processing. In case of non-empty queue the scheduling algorithm will admit another request, a member of the queue, to be processed instantaneously upon removal of the old one. Thus, the processor will switch rapidly among pending requests, giving each a time slice in some cyclic pattern, with due attention to new arrivals. The total processing time, called the service time, required for a request is a random variable which is not known before service starts.

For this process to operate correctly, the computer must have an efficient means of interrupt. At instants of interruptions the current computational status of a request is saved to be restored when processing is resumed.

So far we have represented the system as if there is only one queue awaiting service by the processor. This is not generally true if we consider the queue or queues waiting for the use of input/output (I/O) devices. Since nowhere in this thesis will we discuss the I/O scheduling discipline, we will disregard these queues, but one should keep in mind that a completion of servicing a request by the processor might mean that this given request has reached a point where it requires an I/O process before continuing. Similarly, an arrival may mean a request returning from the I/O system for more processing time.

To consider the scheduling algorithms in more detail, one can define a scheduling algorithm as a set of decision rules determining which user will next be serviced (priority discipline) and how long he will be allowed to use the processor (quantum size) as well as other resources when admitted. These decision rules, which are of primary interest in this research, are always set to satisfy some objective of good service and they appear as part of the supervisor program.

Although storage space can be shared by different programs, usually of the large number of users with pending programs only one or at most a few can be held in the relatively small, expensive main memory at one time, while the rest will be kept on the much less expensive auxiliary storage. The processor can serve a request if it is main memory resident. As each request reaches the time to be processed, if it is not in the main storage, it will be exchanged from auxiliary memory to main memory with some other program which is a main storage resident. This operation of exchanging programs to and from main memory is called swapping and it needs processor time. In many time sharing systems, swapping and processor operations are overlapped in the sense that while a user's request is being run by the processor, the request that was running previously is transferred from the main memory and the request to be run next is loaded to the main memory, since at least two complete user's

requests must be in the same memory at the same time.

The time the processor spends in scheduling, controlling terminal input and output, and other housekeeping tasks represents a slice of time known as overhead time.

In a time sharing operation mode an extra swapping of requests must be performed, and an extra overhead time is usually spent. As overhead and swap times represent losses of processor time, it is obvious that this mode of operation is less efficient than other modes, in particular those which service a request until completion. However, the discussion here is not whether or not time sharing is a proper mode of operation basing the evaluation on hardware efficiency, but it is towards developing the most economical, efficient time sharing system.

Decisions are made not only for the many parameters concerning the hardware configuration (number of terminals, size and speed of the main memory, size and number and speed of the auxiliary memory, number of processors, etc.), but it is for adjustment of the software variables which, in fact, play an integral part in the behavior of this system. In the programmed scheduling algorithm the size of the quantum and many other parameters of the priority discipline are certainly important factors in designing the time sharing system. Also, other variables as those concerned with the interrupt procedure and the core allocation scheme, which direct the detailed

operation of swapping, might have substantial effect upon system performance.

Performance Measures

In the previous discussion we discussed system performance and efficiency, but we did not define them precisely. In trying to find a suitable way to measure the performance of the system, one can see that no single measure can suffice and usually one has to judge the system by two or more diverse criteria, such as good use of resources and minimal delays of user requests. Thus, from the user's point of view, the average response time is the single most important performance measure. A typical objective is to provide a rapid response time to the user who has a short request to be processed. As viewed from the hardware aspects, performance measures for the system resources include overhead time, swap time and hardware utilization, in particular, processor usage. Computer efficiency can be measured by the rate of processing requests, the throughput rate. One can easily see that some of these objectives seriously conflict among each other. Thus the critical question that faces the designer of such systems is to decide precisely what trade off to make among the conflicting objectives.

In most of the time sharing systems the population of users consists of "interactive" and "background" users. The interactive

user interacts with the computer through a terminal, frequently a teletypewriter, in a conversational way. Since interactive users are sensitive to delay, a major objective should be fast response time to those users. Background users are those who are served in a "batch" way. The background workload is characterized in this system by its continual availability with far less demanding response time requirements. The interactive user's requests will enter a queue upon arrival awaiting to be served by the processor while the background requests will only be served when the queue of interactive users is empty. In this case the processor idle time is eliminated, since the processor will serve the background program when the queue is empty; that is, the processor is idle when it does not perform any operation. However, if the idle time is defined as the time the processor spends in swapping and overhead tasks, then one can measure the processor usage as the percentage of time the processor spends in performing users requests compared with the time the processor performs swapping and overhead tasks.

Models for Time-Sharing Systems

Since Scherr (1967) reported as the most important result of his research, that a time sharing system and its users can be successfully modeled and that good accuracy can be obtained from a fairly simple model when compared to the actual system, a large

number of simulation models as well as mathematical models have appeared in the literature of time sharing. These two types of models have been used to solve the many problems of time sharing systems. Following, a brief discussion of the properties of each type of these models is given.

Simulation Models

Simulation techniques have the advantage that no matter how complicated the system is, a great deal can be cast into a model without great distortion. Comprehensive simulation models (Fine and McIssac, 1966; Nielsen, 1967; O'Connor, 1965; Scherr, 1967) for the time sharing systems which contain many details about the hardware configuration, beside details concerning the logic of system operations, have been developed to predict the many parameters of such systems. Examples of these parameters are hardware utilization (processor, disk, etc.), average response time, queue length, number of active terminals, the distribution of the response time, and so on. However, in no sense does this imply that these models are entirely satisfactory, but it does imply that designers are no longer limited to subjective decisions, which showed in the past that in many cases the results were so diverse in effectiveness, or to mathematical models which usually lack the level of detail necessary to handle some of the features of the system. Simulation does not choose its

own variation during runs and come up with an optimal solution. It only describes the behavior of a specified model. That is, it determines how a particular configuration will react in a particular environment. In order to improve the system one should analyze the results of a certain model and decide which and how parameter values may be changed. Modification of these parameters is not a straightforward matter and is done by the human designer, who is still the feedback element in the design loop.

A large number of simulation runs need to be performed in order to test the effectiveness of proposed improvements. A modified model might or might not have the intended effect. For these reasons, studies of this nature can become very time consuming, especially because of the enormous number of parameters in such models, unless parameter selection and variation are carefully limited. It is not a small problem to determine which are the major variations that affect the system. The biggest problem in simulation modeling is to retain all essential details and remove the nonessential features (Seaman, 1966). As reported by Nielsen (1967), in some cases the development of a suitable model might require as much effort and be as costly as developing the time sharing itself. Thus, simulation is necessary only for those portions of the system that are not mathematically analyzable.

Mathematical Models

Mathematical models may lack the richness of descriptive power and oversimplify the practical situation. However, they achieve high degrees of precision if one applies the mathematical results critically. Scherr (1967) developed a very simple Markov model to represent a system like the compatible time sharing system (CTSS) developed at MIT. Many features that would seem essential to operation of actual time sharing systems are not present in this model, swapping, quantum size, and priority scheduling. Moreover, the distribution of arrival time and service time are assumed to be exponential. The purpose of this model is to compare its prediction for the mean response time as a function of the number of user's requests in the system with that of his sophisticated simulation model and with the actual CTSS data, which were taken by a program written to run as part of the CTSS supervisory program.

Figure 1.1 is taken from Scherr's paper, to illustrate the accuracy of the Markovian model in predicting the mean response time. The fact that many details are omitted and yet the model still accurately predicts mean response time is amazing. Since this striking result was published, a large number of mathematical models have been analyzed in order to predict the mean values of some of the time sharing parameters. Examples of these parameters are, the

mean number of active users at any time; i.e., the number of requests in the system either waiting in the queue or being served by the processor, the mean response time conditioned on the length of service required by a given request, where the response time is defined as the total waiting time of the request in the system plus its service time. In these models, considerable attention is directed towards the relation of these parameters and the quantum size, since the quantum size is a design parameter of main interest.

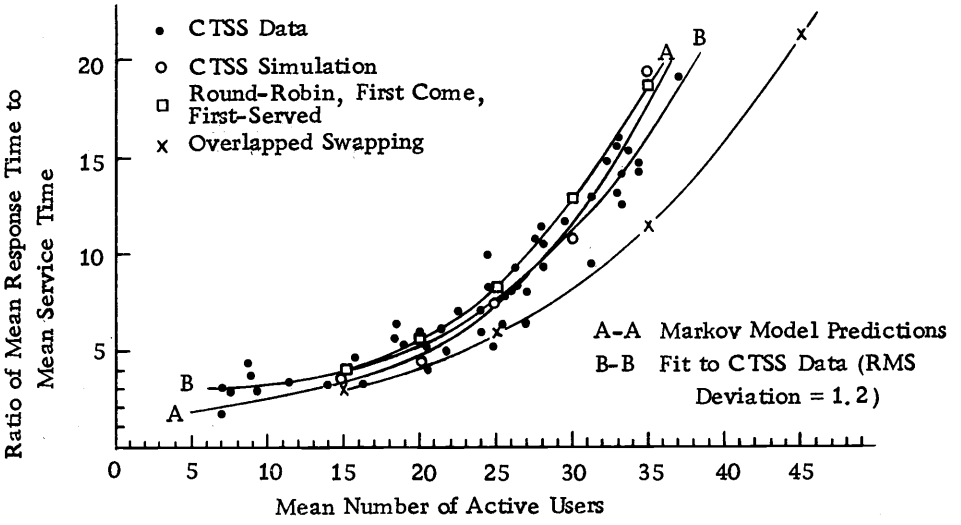

Figure 1.1. Ratio of mean response time to mean length of service required per request vs. mean number of active users in the system.

The usual approach taken in preparing mathematical models is to treat them as a queuing system. In this system, a user's request joins some queue upon arrival and a member of the queue chosen by certain priority discipline (scheduling algorithm) will enter the

service facility (the processor) for, at most, one quantum of service, if it needs more service it will be "fed back" to the system of queues, otherwise it will depart. The system will be defined at any instant by a description of arrival mechanism, the service required from the service facility, the nature of the service facility, and the priority discipline, to which the selection of service requests from the system of users is determined and the condition under which requests are fed back to the system of queues.

Since the priority discipline is of major importance, we will discuss two basic time sharing priority disciplines together with other priority discipline which more properly apply to batch-processing environment. This is in order to measure the performance of the time sharing system as compared with performance in the more efficient but less flexible batch processing environment.

Priority Discipline

The most common priority discipline for the time sharing system is known as a round robin, (RR) (McCarthy et al., 1963). As shown in Figure 1.2, a new arriving request joins the end of the queue, and waits in line. The server allows the first request in line to be processed for at most one quantum, q seconds; if the requested service is completed during this quantum then it leaves the system; otherwise, it is removed from the service facility and cycled

back to the end of the queue to await another turn to continue processing from where it is interrupted. This procedure will continue for all requests in the queue. Each request may have to cycle as many times as needed to complete its total service requirement.

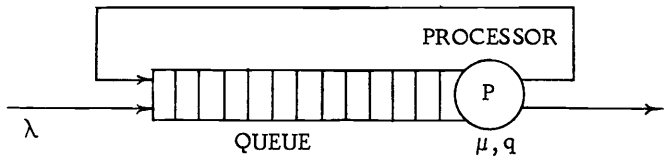

Figure 1.2. Round-robin discipline.

The second major type of time sharing priority discipline is the multiple level feedback (FB_N) (Corbato, 1963), where N is the number of levels. As shown in Figure 1.3, the system might be viewed as consisting of multiple queue levels. New arrivals are put in queue level 1. A request at the service point at any given queue level will not be served unless all lower level queues are empty. Thus, immediately after a request has received service, the next request serviced will be the one at the service point of the lowest level, non-empty queue. This request will be given, at most, a quantum of service as in the round robin discipline. If more service is needed, then the request is subsequently placed at the end of the next higher queue; otherwise, it leaves the system. If the number of

levels, N, is finite, a request at the service point of the Nth queue will be served in some discipline (specified by the model) if all the (N-1)th lower queues are empty.

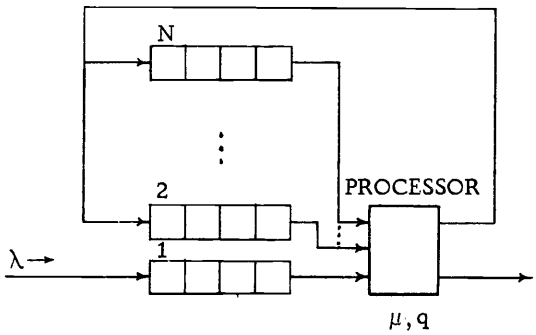

Figure 1.3. FB_N discipline.

The first come first served, (FCFS) (Saaty, 1961) priority discipline which is commonly used in batch processing will be used as a standard reference to evaluate a time sharing priority discipline. An arriving request joins the end of a single queue. The processor services requests, waiting in this queue until completion in the order of their arrival.

The shortest-job first (SJF) (Phipps, 1956) is also discussed here for the same purpose as FCFS. The running time of each request is supposed to be known upon its arrival. Each time the service facility services a request, the queue is inspected. If there is a non-empty queue, the member of the queue that requires the least

processing time will be admitted for service until completion; otherwise, the processor will remain idle (or allowed to serve a background request) until a request arrives which will immediately start service.

Since an efficient assignment of priorities might reduce the costly delay without any increase in system facilities (Cobham, 1954), a careful discussion about the nature of the four priority disciplines follows.

The RR discipline uses both processing time required by a request and its arrival time to make implicit priority decisions. Thus this discipline includes the advantages of a first come first served discipline, as well as, shortest job first served discipline. The benefits of RR over FCFS increase with the uncertainty of service time of a request. The variance of the distribution of request sizes may be taken as a measure of this uncertainty. The basic disadvantage of the RR discipline consists of the swapping of those requests which need more than one quantum of service.

The FB_N discipline has the same advantages and disadvantages as the RR discipline except that the former favors shorter requests and discriminates explicitly against long requests, which are identified by past service. The choice between the RR and FB_N priority disciplines is determined basically by how much one wants to favor short requests, since the two disciplines involve the same amount of

swapping and have the same mean response time.

The FCFS discipline is considered the optimal non-time sharing priority discipline in case the service time required by a request is not known before service. This is because FCFS avoids swapping time and because the response time of a request is less than any other system under the same assumptions. Although this discipline seems to be the most fair one, in fact, it favors the long requests while it safeguards against excessive waiting time and controls the variance of waiting.

The SJFS discipline discriminates explicitly against long requests on the basis of a-priori information on the service time required by each request. It has the advantage that the mean response time in the system is less than any other system including FCFS, while long requests suffer more in SJFS than in FCFS. In fact, the reduction of the mean response time of a request is at the expense of an increase in its variance. Although it is usually difficult to expect accurate advance knowledge of exact running times, it is encouraging to know (Schrage and Miller, 1965) that even with partial indications of service time, significant improvements in mean response time are possible.

User Behavior

Although a proper choice of priority discipline may be done to

improve a system performance, a user may attempt to defeat such systems. For example, if a priority rule places a heavy penalty on the user of a certain type of request (e.g., a long request), the user will try to arrange his request (e.g., divide his request) in such a way that he appears to the system to be a high priority user so he can obtain a better service. Thus, in choosing the priority discipline, the designer should take into consideration the countermeasures available to the users of such discipline.

The four priority disciplines discussed above are the basic disciplines of most of the scheduling procedures that have been considered in the past few years. A number of mathematical models which appeared in the literature of time sharing using these priority disciplines, their modifications or combinations of two will be presented later. In summary all existing models give high priority to the small request either in an explicit or implicit way. This is usually done in order to reduce the overall mean waiting time of the requests in the system. However, this objective should not be met by excessive sacrifice of other desirable properties such as, reduction of swap and overhead times (good use of the processor), especially when this objective loses its practical meaning. A priority discipline which depends on the state of the system would control the degree of favor given to a request in order to avoid the excessive sacrifice to fulfill one objective at the expense of others. This paper is concerned with

the studies of models considering such discipline.

The presentation of the material is as follows. In Chapter II a survey on a number of existing mathematical models is summarized. Chapter III presents the proposed model, its analysis and different techniques developed in order to optimize system performance. In Chapter IV these concepts are illustrated with some numerical examples together with concluding remarks.

II. MATHEMATICAL MODELS

This chapter is a survey of a number of mathematical models which have recently appeared in the literature of time sharing.

The models considered here have a number of assumptions in common. In particular, requests are generated by either a finite or infinite group of users who have direct access to one computer with one processor. Arrival and processing times of a user's request are assumed to be independent random variables. They are also assumed to be independent of the values considered for other requests by the same user or by any other user. Moreover, most of the models require Markovian assumptions for both arrival and service times, which means that the interarrival and service times are assumed to be exponential or geometric random variables, depending on whether the model is analyzed in continuous or discrete time, respectively. According to Scherr (1967) this can represent the real world fairly well in situations where a number of diverse users exist.

Although swapping is the principle bottleneck to efficient operation in time sharing systems, the assumption of zero swap and overhead times is adopted in most of these models. This assumption weakens the mathematical results; however, it is used for the convenience of the analysis. In systems where the swapping of one request overlaps the execution of another, the assumption of zero swap

time is less serious. One can consider the swap time to be represented as a part of the quantum. However, this will change the service time distribution. Or one may think of the zero swap and overhead times models as ideal models in the sense that non-zero swap or overhead times will be at the expense of the accuracy of the predicted performance.

The models are presented in five groups. The first four groups are models where their priority disciplines are: RR, FB_N, FCFS and SJFS, respectively. We will refer to the models which belong to one of the first two groups as quantum controlled models. The fifth group presents some quantum controlled models where the priority disciplines are modifications of those discussed above.

Round Robin (RR) Models

Under the round robin discipline the models are broken into two groups; (a) models where the number of users at any time is assumed to be finite (N terminals), and (b) models with an infinite population of sources.

Scherr (1967) represented the time sharing system as a simple continuous-time Markov process with (N+1) states (N is the number of terminals). In order to do that he assumes that the human response time and the service time are both independently and exponentially distributed. The mean human response time is assumed to

equal to $1/\lambda$ sec. and the mean service time per request is $1/\mu$ sec. The processor is considered as switching from program to program at an infinite rate. In other words if the quantum size is q seconds, then this is the case when $q \rightarrow 0$. For the single processor case, Scherr obtains expressions for the average number of users waiting for service (Q), and the mean response time for all requests (W). He also finds the steady state probabilities for the number of requests in the system, and derives an expression for the mean response time in the case where the processor facility consists of more than one processor. His results for the single processor case are

$$Q = \sum_{i=0}^{N} i\pi_i = \frac{\sum_{i=0}^{N} \frac{(iN)!}{(N-i)!} \rho^i}{\sum_{i=0}^{N} \frac{N!}{(N-i)!} \rho^i} ,$$

and

$$W = \frac{N}{(1-\pi_0)\mu} - \frac{1}{\lambda} ,$$

where

$$\pi_0 = \frac{1}{\sum_{i=0}^{N} \frac{N!}{(N-i)!} \rho^i} ,$$

$$\pi_i = P_r\{i \text{ requests are in the system either receiving or awaiting service}\},\ i = 0, \ldots, N,$$

and

$$\rho = \frac{\lambda}{\mu}.$$

These results are the same in the case where one assumes that each request is run until completion (FCFS discipline). This is due to the assumption that no time is lost by the processor for swap or overhead tasks and that the time distributions are exponential.

Greenberger (1966) analyzes a model with the same assumptions as the previous one except that the quantum size is finite and the swap time is a constant and equal to τ seconds. He obtains an approximation which can be used to find the mean response time conditioned on the service time required (W(kq)), (kq, is the service time required for service in seconds),

$$W(kq) = \frac{k}{\mu'}\left[1-e^{-\mu q+\mu\tau}\right]\left[\frac{N}{1-\pi'_0} - \frac{\mu}{\lambda}\right] + \left[\frac{1-\pi'_0(1+N\frac{\lambda}{\mu})}{1-\pi'_0}\right]\left[\frac{S_2-\tau^2}{2(S_1+\tau)} - S_1\right],$$

where

$$\frac{1}{\mu'} = \frac{1}{\mu} + \frac{\tau}{1-e^{-\mu q}},$$

$$\pi'_0 = \left[\sum_{i=0}^{N} \frac{N!}{(N-i)!} \rho'^{i} \right]^{-1} ,$$

$$S_1 = \frac{1}{\mu} (1 - e^{-\mu q}) ,$$

$$S_2 = \frac{2}{\mu} (S_1 - q e^{-\mu q}) ,$$

$$\rho' = \frac{\lambda}{\mu'} ,$$

and where μ and λ are as defined in the previous model.

Krishnamoorthi and Wood (1966) study a model where both human response time and service time possess exponential distribution. The length of the quantum is assumed to be composed of two parts; an operation part which is a random variable and a constant swap time part. The overhead time is considered to be zero. As the process is non-Markovian, an imbedded Markov chain is analyzed by considering the instants of completion of quantum of service as regeneration points. Expressions for the mean and the variance for the number of active users in the system are derived. Also, expressions for the mean and the variance of the cycle time are found, where the cycle time is defined as the period between two consequtive quanta of service on one request (the cycle time is defined only for a request which needs more than one quantum of service). Using the concept of cycle time, an expression for the mean response time

conditioned on the length of service required, which is not fully determined, is given.

Chang (1966) treats the quantum size as a random variable. He assumes that the flow of requests for service from each input source constitute a Poisson process. If λ_i is the input density of the Poisson process for source i, $i = 1, 2, \ldots, N$, then the total input to the processor is a Poisson process with density λ, where

$$\lambda = \sum_{i=1}^{N} \lambda_i .$$

The service time distribution, B(t), is assumed to be arbitrary with the rth moment defined as

$$a_r = \int_0^{\infty} t^r dB(t).$$

A relation between the service time distribution and the distribution of the quantum size, X(t), is derived. The model is analyzed by looking at the system after the completion of the nth quantum. If ξ_n is the queue size in the system immediately after the completion of the nth quantum, $\{\xi_n\}_{n=0}^{\infty}$ form a Markov chain. By solving for the generating function for the queue size immediately after a departure of a request, the average number of users in the system, Q,

is obtained as

$$Q = \frac{\lambda^2 b_2 - 2\lambda b_2(1-\lambda b_1)}{2(a-\lambda b_1)},$$

where

$$b_r = \int_0^\infty t^r dX(t),$$

and

$$a = P_r \{\text{a request is completely served after a quantum of service}\}.$$

The mean response time, W, is given as

$$W = Q/\lambda = \frac{\lambda b_2 + 2b_1(1-\lambda b_2)}{2(a-\lambda b_1)},$$

and the second moment of response time, W_2, is given as

$$W_2 = \frac{a^2 - 2a}{6(a-\lambda b_1)^2[a^2 - a(2+\lambda b_1)+\lambda b_1]}$$

$$\times \{2a[6\lambda b_1^3 - 6b_1^2 - 2\lambda b_1 b_2 + \lambda b_3]$$

$$- [12\lambda b_1^3 - 12 b_1^2 - 6\lambda b_1 b_2 + 2\lambda^2 b_1 b_3 - 3\lambda^2 b_2^2]\}.$$

When the number of users is infinite, a model is assumed to represent a large processing system which has an access to a huge

number of terminals. In such a system, usually one can see a large number of idle terminals at the same time even at the heaviest traffic.

Kleinrock (1964) considers the discrete time model where time is quantized into segments each q seconds in length. At the end of each time interval a new request arrives in the system with probability λq. The service time of a request is assumed to have a geometric distribution such that for $\sigma < 1$

$$S_k = (1-\sigma)\sigma^{n-1}, \quad k = 1, 2, 3, \ldots, \tag{2.1}$$

where S_k is the probability that a request's service time is exactly k time intervals long, i.e., its service time is kq seconds. For such systems, he shows that the expected value, $W(kq)$, of the response time in the round-robin system for a request whose service time is kq seconds, is

$$W(kq) = \frac{kq}{1-\rho} - \frac{\lambda q^2}{1-\rho}\left[1 + \frac{(1-\sigma a)(1-a^{k-1})}{(1-\sigma)^2(1-\rho)}\right],$$

where

$$a = \sigma + kq, \quad \text{and} \quad \rho = \frac{\lambda q}{1-\sigma}.$$

Furthermore, the expected number, Q, of customers in the system is given by

$$Q = \frac{\rho\sigma}{1-\rho} .$$

By keeping the average service time and the average arrival rate and letting $q \rightarrow 0$ a model which is referred to as processor-shared model is analyzed (Kleinrock, 1967).

Coffman and Kleinrock (1968a) considered a continuous round-robin model in which the interarrival and service times of a request are assumed to be exponential. For such a system, they define the "quantum-service" distribution as follows:

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

Then the mean waiting time in the continuous RR system of a request requiring t seconds of service is shown to be

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

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

where

λ is the mean arrival rate,

μ is the mean service rate,

$\rho = \lambda/\mu$,

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

R is the smallest integer such that $kq > t$,

and

$E_1(\tau^2)$ is the second moment of the quantum-service distribution and is given by

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

Multiple Level Feedback (FB_N) Models

Coffman and Kleinrock (1968a) study the case where N is assumed to be finite. Requests at the Nth level are served a quantum at a time until completion. An arrival to a lower level during the servicing of the Nth level request will preempt this request after it has completed the quantum-service in progress. Also, interarrival and service times are assumed to be independently and exponentially distributed. For this model, they show that a request requiring t seconds of service in the FB_N system has an expected waiting time in the system of

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

$$1 \leq k \leq N-1,$$

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

$$k \geq N,$$

where k is the smallest integer such that $kq > t$ where $E_k(\tau)$ and $E_k(\tau^2)$ are defined as

$$E_k(\tau) = \frac{1}{\mu}(1-e^{-\mu kq}),$$

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

and where

$$\gamma_k = \frac{e^{-\mu kq}}{1-e^{-\mu q}},$$

λ is the mean arrival rate and μ is the mean service rate.

Schrage (1967) has provided a general analysis of the FB_N model in the case $N = \infty$. In particular, the Laplace transform of the waiting time distribution is found under the assumption of arbitrary quantum size for each level.

First-Come-First-Served (FCFS) Models

On the assumption of Poisson arrival and exponential service time, this is the conventional (M/M/1) queue. Kleinrock (1964)

considers the discrete time case where the time is quantized with segments each q seconds in length and a new arrival arrives in the system at the end of a time interval with probability λq. The service time of an arriving request is assumed to be geometric such as that in (2.1). For such systems he shows that the expected value, $W(kq)$ of the response time in the strict first-come-first-served system for a request whose service time is kq seconds is

$$W(kq) = \frac{qE_r}{1-\sigma} + kq ,$$

where

$$E_r = \frac{\rho\sigma}{1-\rho} ,$$

$$\rho = \frac{\lambda q}{1-\sigma} ,$$

and σ is defined as the probability that a request having just operated for one quantum requires at least one more in order to complete its service.

Shortest-Job-First-Served (SJFS) Model

Phipps (1956) analyzes this model under the assumption of Poisson arrival and exponential service time where the service time of a request is known at the time of arrival. He derives the mean waiting time in the queue of a unit whose service requirement is t

seconds,

$$W(t) = \frac{\rho(\frac{1}{\mu})}{1-\frac{\lambda}{\mu}[1-e^{-\mu t}(1+\mu t)]^2} ,$$

where

λ is the mean arrival rate,

μ is the mean service rate,

and

$$\rho = \frac{\lambda}{\mu} .$$

Other Priority Models

Coffman (1968) analyzes two models with a quantum allocation procedure that depends not directly on the state of the system but rather on changes to the state of the system. In particular, whenever the number in the system increases, a request, if any, receiving a quantum at that time is given another quantum, if it is required. Thus a request continues to receive service until it either completes its service or receives a quantum during which there is no arrival. In the latter case the request is returned to the queue awaiting another turn of service according to some discipline. In the first model, Coffman considers the round robin service discipline and analyzes the model in discrete time assuming that both arrival and service times of requests have geometric distributions. He finds

that the mean waiting time for a program requiring k quanta of service is

$$W(k) = \frac{k\rho q}{1-\rho} - \rho q - \frac{\lambda \rho q}{1-\rho}\left[1+\frac{(1-\sigma\alpha)(1-\alpha^{k-1})}{(1-\sigma)^2(1-\rho)}\right],$$

where

$$\alpha = \lambda + \sigma,$$

$$\rho = \frac{\lambda}{1-\sigma},$$

λ is the probability that an arrival occurs in a quantum q, and σ is the probability that a request having just operated for one quantum requires at least one more in order to be completed. In the second model Coffman discusses the case where the quantum allocation procedure described above is applied to the basic FB_2 model, "forground-background" model. He finds the mean waiting time in the queue of those requests requiring only one quantum of service, W_F, and the mean waiting time in the queue of requests requiring more than one quantum, W_B, as

$$W_F = \frac{\lambda \sigma^2 q}{(1-\lambda)(1-\sigma)^2},$$

and

$$W_B = \frac{\lambda \sigma q (1-\sigma)^2}{(1-\lambda)[1-\lambda/(1-\sigma)]} + \frac{\lambda q}{1-\lambda},$$

where σ, q, and λ are defined as in the first model.

Other types of models in which external priorities are assigned to arriving requests are also analyzed for the basic quantum controlled priority disciplines. Kleinrock (1967) discusses a round robin model with external priorities where a higher priority request is allocated a larger quantum than a lower priority request. Coffman and Kleinrock (1968a) discuss the FB_N model with priorities. Requests arrive at any queue level according to assigned external priority. As in the basic FB_N model, the lowest level, non-empty queue is chosen for service, and service is allocated q seconds at a time with requests requiring more service moving up level by level.

III. OPTIMIZATION TECHNIQUES FOR TIME SHARING SYSTEMS

From the previous chapter, one can see that analytical studies of time sharing have tended to derive the expected number of active users and the mean response time conditioned on the length of service required by a request. Based upon these derivations, a conclusion is generally given about the efficiency of the system under analysis. This leaves the matter of optimum quantum size ambiguous. For this matter the approach of costing the model has been used (Greenberger, 1965) and an expression to compare the cost implications of different choices of quantum size has been derived. From the derivations, one can find an optimal decision procedure for the quantum size; however, this decision procedure is rigid in the sense that the size of the quantum does not depend on the state of the system (e.g., the number of requests in the queue).

A more flexible decision procedure allows the size of the quantum to vary with the state of the system. Such models are the main concern of this chapter. When the number of active users is relatively small, one needs to lengthen the quantum so that the response time of a common request[1] should not be less than the human reaction

---

[1] In later discussions, the response time of a common request will be replaced by the cycle time of any request in the system where the cycle time is rougly defined as the time required to execute one round robin service through all active users.

time; a common request is defined as a request which needs service for, at most, one quantum size. Conversely, if the system is heavily congested, one might wish to shorten the quantum so that the response time of a common request should not exceed the human reaction time of this request. In other words, we need to discuss the possibility of constructing a decision procedure for the quantum size which forces the response time of a common request to converge to the human reaction time of such a request. The motivation of such a system is to reduce swap and overhead times, especially when the system is not heavily congested. In order to decrease the effect of degrading the system performance at heavy load time, the size of the quantum should never be less than the mean service time of the common request. With these concepts, models for time sharing systems are devised and different techniques are discussed for optimizing such models.

In these models, the population of users is considered to consist of interactive and background users. A background request will be served only if there is no interactive request waiting in the system, and will be pre-empted from the service facility upon the arrival of an interactive request. The analysis will be carried only for the interactive requests while no concern will be given to the background requests.

The scheduling discipline for interactive requests[2] is a round robin with variable quanta. The basic parameter of our system is the cycle time and the control parameter is the quantum size which is allocated dynamically in such a way as to regulate the cycle time in the system. An optimal decision procedure for the quantum size is considered to force the cycle time to approach the human response time of the common request. In our studies, it is assumed that the first and second moments of the human response time, for the common request, are given (through observation and statistical analysis).

Before presenting our models, it is necessary to distinguish between the quantum size, which is a fixed length of allocated service and the quantum service, which is a random variable equal to the actual amount of time a request spends in the service facility. In addition, a precise definition of the cycle time must be given. It is convenient to make the definition with reference to a given request known as the "tagged" request. At the instant of completion of the quantum service on the tagged request, three cases may prevail: (1) the tagged request needs at least one more quantum of service, (2) the tagged request completes its service and the queue of active users is not empty, or (3) the tagged request completes its service

---

[2] In the following discussion the term interactive will be dropped. Requests should be understood as interactive requests.

and the queue of active users is empty. In the three cases the instant of beginning a new cycle coincides with the instant of completion of a quantum service on the tagged request. The cycle time is defined as the interval until the tagged request receives one more quantum service where the tagged request in (2) is the request at the end of the queue at the instant when the old tagged request completes its service. In (3) the beginning of a new cycle is the instant when the first arrival completes one quantum of service, if it needs more service after this quantum. It should be noticed here that no cycle time is defined otherwise. Thus at the beginning of a cycle, the cycle time, T_s, is the time it would take to give each of the requests, the active users awaiting service at this instant, one quantum service. From our definition of the cycle time there is at least one such request.

The first and the principal model discussed in this thesis is a state dependent, cycle oriented model which can be specified in terms of the following assumptions.

1. There is only one processor in the system.
2. The number of active users, $\xi(t)$ at any time, t, is considered to be finite and at most equal N.
3. For the rth user; (a) Let the time interval between completing his $(i-1)$th request and the arrival of his ith request be $\theta_{r,i}$. Assume that the sets of random variables

$\{\theta_{r,i}\}_{i=1}^{\infty}$ for $r = 1, \ldots, N$ are mutually independent and that the sequence $\{\theta_{r,i}\}_{i=1}^{\infty}$ is a sequence of independently and identically distributed random variables with the distribution function $A(t)$ given by

$$A(t) = \begin{cases} 1-e^{-\lambda t} & t \geq 0 \quad \lambda > 0, \\ 0 & t < 0, \end{cases} \tag{3.1}$$

with a mean of $1/\lambda$ seconds.

(b) Let the service time of his ith request, i.e., the total time required to serve his ith request until completion, be $\gamma_{r,i}$. The sets of random variables $\{\gamma_{r,i}\}_{i=1}^{\infty}$ for $r = 1, \ldots, N$ be assumed to be mutually independent and identically distributed random variables with the distribution function $B(\tau)$ given by

$$B(\tau) = \begin{cases} 1-e^{-\mu \tau} & \tau \geq 0 \quad \mu > 0, \\ 0 & \tau < 0, \end{cases} \tag{3.2}$$

with a mean of $1/\mu$ seconds. The θ's and γ's are assumed to be mutually independent.

4. The service discipline is round robin where the quantum size is determined at the beginning of each cycle as a function of the number of active users awaiting service. If at this instant there are i requests then q_i seconds will be

allocated to each one of the i requests. If the service required by any of these requests is completed during the q_i seconds from the time this request is admitted for service, then it will depart at the instant of completion, otherwise it will join the end of the queue after exactly q_i seconds. Subsequent cycles are then determined in the same way.

5. The swap time is assumed to be a fixed value and equal to τ while the overhead time is assumed to be zero. τ may represent the average time required to bring a request in and out of the main memory. The effect of τ is to lengthen the service required by a request and hence change the distribution of service time.

Quantum Service Distribution

The distribution of the quantum service, which is the actual amount of time a request spends in the service facility once it is admitted, can easily be derived, from (3.2) as

$$F_i(x) = \begin{cases} 0 & x < \tau, \\ 1-e^{-\mu(x-\tau)} & \tau \leq x < q_i + \tau \\ 1 & x \geq q_i + \tau, \end{cases} \qquad (3.3)$$

where q_i is the quantum size allocated to this quantum service.

Because of the memoryless property of the exponential distribution one should keep in mind that (3.3) holds, no matter how many times a request admitted for service has received previous service.

Let $X(q_i)$ denote the quantum service random variable for which a length of time q_i is allocated.

$$X(q_i) = \begin{cases} \tau + q_i & \text{with probability } e^{-\mu q_i}, \\ R(q_i) & \text{with probability } 1-e^{-\mu q_i}, \end{cases} \tag{3.4}$$

where $R(q_i)$ is a random variable with density function $f_{R(q_i)}(x)$ given by

$$f_{R(q_i)}(x) = \frac{\mu e^{-\mu(x-\tau)}}{1-e^{-\mu q_i}}, \qquad \tau \leq x \leq q_i + \tau . \tag{3.5}$$

The expected value of the quantum service is

$$E(X(q_i)) = \int_0^{q_i} x\,dF_i(x) = (q_i+\tau)e^{-\mu q_i} + (1-e^{-\mu q_i})\int_0^{q_i} \frac{x\mu e^{-\mu(x-\tau)}}{1-e^{-\mu q_i}} \tag{3.6}$$

$$= \tau + \frac{1-e^{-\mu q_i}}{\mu} = a_{1,i},$$

and the second moment of the quantum service is

$$(3.7)\qquad E(X^2(q_i)) = \int_0^\infty x^2 dF_i(x) = \frac{2}{\mu^2}(1-e^{-\mu q_i})(\frac{1}{\mu}+\tau) - 2q_i\frac{e^{-\mu q_i}}{\mu}$$

$$= a_{2,i}.$$

Thus its variance is

$$(3.8)\qquad Var(X(q_i)) = (a_{2,i}-a_{1,i}^2) = \frac{1}{\mu^2} - \frac{2q_i e^{-\mu q_i}}{\mu} - \frac{e^{-2\mu q_i}}{\mu^2}.$$

Queue Size Distribution

Let $\phi(t)$ denote the number of requests awaiting service, which will be denoted as the length of the queue, at time t. Although both arrival and service times have the exponential distribution, this process is not a Markovian process as long as the swap time τ is greater than zero. If we investigate the system at these instants when a new cycle begins, one can justify that the cycle time is independent of the time t but is dependent on the length of the queue at the time the cycle starts. Thus if t_n is the instant when the nth cycle begins for $n = 0, 1, 2, \ldots$ and $\phi_n = \phi(t_n)$ is the queue length at time t_n and since the future development of the queue does not depend on what happens before t_n, then $\{t_n\}_{n=0}^{\infty}$ is a sequence of regeneration points for the stochastic process $\{\phi(t); t \geq 0\}$ and the sequence $\{\phi_n\}_{n=0}^{\infty}$ is a discrete time Markov

chain with state space $\{1, 2, \ldots, N\}$. This chain is a time-homogeneous Markov chain and hence the elements of its transition probability matrix, $A = (a_{ij})$ are independent of the time t, where,

$$(3.9) \qquad a_{ij} = P_r\{\phi_{n+1} = j \mid \phi_n = i\}, \quad i, j = 1, \ldots, N .$$

To find the elements a_{ij}, let $\phi_{n+1} > 1$, then $t_{n+1} - t_n$ is the time the processor spends to give one quantum service to each of the ϕ_n requests. If r_m is the instant the processor completes the mth quantum service, $m = 0, 1, 2, \ldots,$ and if $\xi_n = \xi(r_m)$ is the queue length at time r_m, then the sequence $\{\xi_m\}_{m=0}^{\infty}$ is a Markov chain with state space $\{0, 1, \ldots, N\}$. At an instant t_n, the instant of starting the nth cycle, if the queue length is greater than one then there exists an integer $k = k(n)$ such that $t_n = r_k$ (this is from the definition of the cycle time). Thus we can write the following expression

$$(3.10) \qquad a_{ij} = P_r\{\phi_{n+1} = j \mid \phi_n = i\} = P_r\{\xi_{k+i} = j \mid \xi_k = i\}, \quad \text{for } i, j \geq 2,$$

The case when $\phi_{n+1} = 1$ we have

$$(3.11) \qquad a_{i1} = P_r\{\phi_{n+1} = 1 \mid \phi_n = i\} = P_r\{\xi_{k+i} = 1 \mid \xi_k = i\} + P_r\{\xi_{k+1} = 0 \mid \xi_k = i\}.$$

Following Krishnamoorthi and Wood (1966) the elements of the transition probability matrix $P(q_\ell) = (p_{ij}(q_\ell))$, denoted

$$(3.12) \qquad p_{ij}(q_\ell) = P_r\{\xi_{k+1} = j \,|\, \xi_k = i, q_\ell\} \qquad i, j = 0, 1, \ldots, N,$$

where q_ℓ is the quantum allocated to serve a request admitted at the $k+1$ quantum service is

$$(3.13) \qquad p_{ij}(q_\ell) = \begin{cases} 0 & j < i-1, \quad i \geq 1, \\ \int_0^{q_\ell} \mu e^{-\mu q_\ell} P(0, t \,|\, N-i)dt, & j = i-1, \quad i \geq 1, \\ \int_0^{q_\ell} \mu e^{-\mu t} P(j-i+1, t \,|\, N-i)dt & \\ \quad + e^{-\mu q_\ell} P(j-i, q_\ell \,|\, N-i), & j \geq i, \quad i \geq 1, \end{cases}$$

and

$$p_{0j}(q_\ell) = p_{ij}(q_\ell), \qquad \text{for all } j,$$

where

$$(3.14) \qquad P(i, t \,|\, N) = \begin{cases} \binom{N}{i} (a(t))^i (1-a(t))^{N-i} & N \geq i \\ 0 & \text{otherwise} \end{cases}$$

and

$$a(t) = 1 - e^{-\lambda(t-\tau)} .$$

Substituting (3.14) into (3.13) and evaluating the integral we find

$$
(3.15)\qquad p_{ij}(q_\ell) = \begin{cases} 0, & j < i-1, \quad i \geq 1, \\ \dfrac{\mu e^{-\lambda\tau}}{\mu+\lambda}\left(1-e^{-q_\ell(\mu+\lambda)}\right), & j = i-1, \quad i \geq 1, \\ \left(\mu\binom{N-i}{j-i+1}e^{-\lambda\tau(N-j-1)}\right) & \\ \quad \times \displaystyle\sum_{k=0}^{j-i+1} \binom{j-i+1}{k} \frac{(-1)^k e^{-\lambda\tau k}\left(1-e^{-q_\ell(\mu+\lambda(N-j-1+k))}\right)}{\mu+\lambda(N-j-1+k)} & \\ \quad + e^{-q_\ell\mu}\binom{N-i}{j-i}\left(1-e^{-\lambda(q_\ell+\tau)}\right)^{j-i}\left(e^{-\lambda(q_\ell+\tau)}\right)^{N-j}, & \\ & j \geq i, \quad i \geq 1. \end{cases}
$$

From (3.10), (3.11), and the property of Markov chain we have

$$
(3.16)\qquad a_{ij} = \begin{cases} p_{ij}^{(i)}(q_i), & j \geq 2, \\ p_{i0}^{(i)}(q_i) + p_{i1}^{(i)}(q_i), & j = 1, \end{cases}
$$

where

$$
(p_{ij}^{(i)}(q_i)) = P^i(q_i) \qquad \text{for} \quad i, j = 1, \ldots, N,
$$

and $P^0(q_0)$ is the identity matrix.

The Markov chain $\{\phi_n\}_{n=0}^{\infty}$ is irreducible, aperiodic and therefore ergodic, since the state space is finite. Thus a limiting distribution exists which is independent of the initial conditions. The limiting distribution, Π, which is a row vector, $(\pi_1, \pi_2, \ldots, \pi_N)$,

can be determined from the relations

$$(3.17)\qquad \sum_{i=1}^{N} \pi_i a_{ij} = \pi_j, \qquad \sum_{i=1}^{N} \pi_i = 1, \qquad \text{and} \qquad \pi_i \geq 0.$$

Solution of (3.17) is assumed to be performed numerically. The vector Π tells about the limiting behavior of the length of the queue at the moments when a new cycle is about to start.

Moments of the Cycle Time

The cycle time, T_s, has the distribution

$$(3.18)\qquad P_r\{T_s \leq x\} = \sum_{i=1}^{N} P_r\{T_s \leq x \mid \phi_n = i\}\, P_r\{\phi_n = i\},$$

where ϕ_n is the queue length when the nth cycle is about to start. Let

$$(3.19)\qquad S_{\phi_n} = \sum_{i=1}^{\phi_n} X_i \qquad \text{for} \quad \phi_n = 1, 2, \ldots, N,$$

where X_i is the ith quantum service since the instant that the current cycle is about to start.

Define,

(3.20) $$G_i(x) = P_r\{S_i \leq x\},$$

where the quantum size q_i is allocated to serve each of the i requests. Thus

$$P_r\{T_s < x\} = \sum_{i=1}^{N} \pi_i G_i(x), \quad x \geq 0 .$$

To find the moments of the cycle time we have

(3.21) $$E(T_s) = \sum_{i=1}^{N} \pi_i E(S_i) = \sum_{i=1}^{N} \pi_i \int_0^{q_i} x dG_i(x)$$

$$= \sum_{i=1}^{N} i\pi_i \int_0^{q_i} x dF_i(x) = \sum_{i=1}^{N} i\pi_i a_{1,i} ,$$

where $a_{1,i}$ is given by (3.6).

Also

(3.22) $$E(T_s^2) = \sum_{i=1}^{N} \pi_i E(S_i^2)$$

$$= \sum_{i=1}^{N} \pi_i [Var(X_i(q_i)) + (iE(X_i(q_i)))^2]$$

$$= \sum_{i=1}^{N} \pi_i [i(a_{2,i} - a_{1,i}^2) + i^2 a_{1,i}^2] ,$$

where $a_{2,i}$ is given by (3.7).

Since the system is inspected at the beginning of a new cycle, the length of the queue is a fixed value at this moment. If it is observed to be equal to i then

$$E(T_s|i) = E(S_i) = ia_{1,i}, \tag{3.23}$$

and

$$E(T_s^2|i) = E(S_i^2) = i(a_{2,i}-a_{1,i}^2) + i^2a_{1,i}^2 . \tag{3.24}$$

One should mention here that the analysis is strongly dependent on the definition of the cycle time. Alternative definitions of this random variable might lead to different results. To demonstrate this fact, the system should be inspected at those instants when a tagged request is about to be admitted for service. On the assumption that this request will return for at least one more quantum service after the current one, the cycle time in this case is defined as the time between this instant and the request's next admission to service. Similar to the previous discussion, one can find the transition probability matrix $B(q_\ell) = (b_{ij}(q_\ell))$ where

$$b_{ij}(q_\ell) = \sum_{k=0}^{N-i-1} \binom{N-i-1}{k} (1-e^{-\lambda q_\ell})^k (e^{-\lambda q_\ell})^{N-i-1+k} p^{(i+k)}_{i+1+k,\,j+1}(q_\ell) \tag{3.25}$$

$$i,j = 0,1,2,\ldots,N-1$$

where $p_{ij}(q_\ell)$ is given by (3.15), $p^{(n)}_{ij}(q_\ell)$ is the (i,j)th element

of the matrix $P^n(q_\ell)$ and P^0 is defined as the identity matrix. Thus the limiting distribution of the length of the queue at the instants of beginning a new cycle, by the definition, is different from that given by the previous one. This implies that the expected length of the queue is not in equilibrium at any time except for those instants at which a Markov chain is imbedded, i.e., the beginning of a new cycle.

In this stochastic situation we want to determine a decision rule to assign the quantum size dependent on the length of the queue according to some cost criteria. The cost function is chosen to be quadratic, not only for the mathematical convenience in part of our discussion but it is also believed that such a choice is proper to measure the performance of the system. To explain that in further detail, the cost function used is $E(T_s-\theta_1)^2$, the expected square difference between the cycle time and the desired cycle time. If T_{s_ℓ} denotes the ℓth cycle time, and if it is possible to find a decision rule such that the cost function tends to zero as ℓ tends to infinity, then T_{s_ℓ} converges to θ_1 in quadratic mean. In fact our cost function serves as an error criterion of a regulator whose purpose is to maintain the system as close as possible to equilibrium state. If the system can achieve equilibrium, at some time, and remain there, we must have

$$\sum_{\ell=0}^{\infty} E(T_{s_\ell} - \theta_1)^2 < \infty, \tag{3.26}$$

which implies that T_{s_ℓ} converges to θ_1 with probability one. Thus, in such a case, we will be minimizing our cost function by reaching asymptotic stability.

A physical interpretation of minimizing $E(T_s - \theta_1)^2$ is, that the cost of a cycle finishing ahead of the desired time is the same as the cost of a longer cycle by an equal time. This is also a desirable property for our performance criteria, since the first case implies inefficiency in using the service facility and the second implies intolerable response time for the users.

For optimizing our model, two approaches have been investigated. The first is the stochastic programming approach and the second is the optimum control system approach.

The Stochastic Programming Approach

To formulate our problem as an optimization problem, we consider the system at those instants when a new cycle begins. As we showed earlier in the chapter we are in one of a finite number of states, N, where the transition from one state to another is governed by the transition probability matrix $A = (a_{ij})$, given by (3.16). There is also a "cost" vector $c = (c_i)$, where c_i is

the cost associated with state i, given as

$$c_i = E(S_i-\theta_1)^2 = E\Big(\sum_{k=1}^{i} X_k(q_i)-\theta_1\Big)^2 \tag{3.27}$$

S_i is the cycle time given that there are i users in the queue when the current cycle starts. By (3.6) and (3.7)

$$c_i = i(a_{2,i}-a_{1,i}^2) + (\theta_1-ia_{1,i})^2. \tag{3.28}$$

A single choice of the quantum size for each state will determine the matrix A and the vector $\underline{c}$ uniquely. Thus one predetermines for each state a finite number of alternatives for the quantum size. If the system is in state i, i.e., the length of the queue is i, one of the predetermined alternatives associated with state i will be selected. This selection is called a decision and the set of decisions for all states is a policy (a decision rule), which can be denoted by a row vector $\underline{q} = (q_1, \ldots, q_N)$. We should mention here that the decision alternatives are assumed to be time-invariant. Although the number of alternatives from each state can be different, we shall assume that the number of alternatives is equal to m in all states.

Our objective is to find the optimal policy, i.e., the policy which minimizes the expected cost, g, per unit time, where

$$g = \sum_{i=1}^{N} \pi_i c_i, \tag{3.29}$$

and π_i is the ith component of the limiting distribution of the system. Such an optimal policy belongs to the class of all policies, R, which contain m^N elements.

Hypothetically, the optimal policy can be found by enumerating all m^N policies and computing (3.29); however, practically speaking, this cannot be done because of the size of m^N.

Howard (1960) provides an iterative algorithm to find the optimal policy within a subclass of the class R. This algorithm is known as the policy iteration method and is composed of two parts: the value determination operation and the policy improvement routine. A brief summary of this algorithm is given below:

The Policy Iteration Method

(a) The value determination operation:

i) choose a starting policy $\{k_0(i)\}$.

ii) set $v_0(N) = 0$ and solve the N following equations for $v_0(1), \ldots, v_0(N-1)$ and g_0,

$$g_0 + v_0(i) = c_i^{k_0(i)} + \sum_j a_{ij}^{k_0(i)} v_0(j), \quad i, j = 1, \ldots, N, \tag{3.30}$$

where g_0 is the expected cost per cycle, $v(i)$ is the relative cost for the system starting from state i instead of state N, $a_{ij}^{k_0(i)}$ is the (i,j) element of the transition probability matrix A, given $k_0(i)$ and $c_i^{k_0(i)}$ is the cost associated with state i given $k_0(i)$.

(b) The policy improvement routine:

iii) find the value $k_1(i)$ which minimizes

$$c_i^{k_1(i)} + \sum_j a_{ij}^{k_1(i)} v_0(j), \tag{3.31}$$

where $v_0(j)$ are those obtained by (ii).

iv) using the new policy, $\{k_1(i)\}$ repeat cycle at (ii) until the process converges.

Howard (1960) shows that the process will converge in a finite number of iterations and that in the iteration cycle, the cost decreases with each successive policy until it attains its minimum. Thus if one starts with a near optimal policy as a starting policy (determined by means of an approximation technique (White, 1969) or by intuition), he may reduce the computational cost appreciably. If there is no a priori reason for selecting a particular initial policy, then it is convenient to start Howard's algorithm from the policy improvement routine with all $v(i) = 0$ for $i = 1, \ldots, N$.

Although Howard's technique reduces the amount of computations to a large extent if we compare it with enumeration methods, its requirements might be excessive to treat large scale problems. For systems with a large number of states the work in step (ii) might be very involved. Also one needs $\frac{(mN^2)(N+1)}{2}$ vector multiplications in order to compute the transition probability matrix A. These complications in computation are negligible compared to the memory required to store the transition matrices $A(q) = (a_{ij}(q))$ and the vectors $\underline{c}(q) = (c_i(q))$.

Linear Programming Technique

An alternative method to Howard's technique is provided by Manne (1960). This method depends on the fact that the limiting probability distribution can be found by means of linear programming. Let $\pi_{ik(i)}$ be the limiting probability of being in state i given decision $k(i)$. One can formulate the problem as finding $\pi_{ik(i)}$ that satisfy,

$$\sum_{k(j)=1}^{m} \pi_{jk(j)} = \sum_{i=1}^{N} \sum_{k(i)=1}^{m} a_{ij}^{k(i)} \pi_{ik(i)}, \qquad i, j = 1, \ldots, N,$$

$$\sum_{i=1}^{N} \sum_{k(i)=1}^{m} \pi_{ik(i)} = 1, \tag{3.32}$$

and

$$\pi_{ik(i)} \geq 0,$$

and that minimize the function,

$$g = \sum_{i=1}^{N} \sum_{k(i)=1}^{m} \pi_{ik(i)} c_i^{k(i)}, \tag{3.33}$$

where $a_{ij}^{k(i)}$ and $c_i^{k(i)}$ are defined as before. Manne showed that the minimal value of g subject to (3.33) can always be achieved by a pure policy, i.e., one in which, for a given state, i, $\pi_{ik(i)} = 0$ except for one value of $k_{(i)}$. Thus solving the above linear programming, with $m \times N$ variables and $N + 1$ constraints, other than the non-negativity conditions, will produce the optimal policy required. Again for large m and N the computations and the memory requirements become excessive.

The main shortcoming in optimizing the principal model is contained in the fact that we are only concerned with those requests which start new cycles and thus ignore all requests served ahead of the tagged requests. The average number of requests in the system may fluctuate all the time except for these particular instants when a new cycle starts. At those instants the system is stable and the type of the queue that any tagged request finds when it starts its cycle is

the same. Thus while we are regulating the cycle for those tagged requests, the effect of our policy on other requests is not clear. For this reason one may think of inspecting the queue at the moments of completion of a quantum service where the decision to allocate the size of the quantum for the next request admitted is made at these instants of inspections. However, this modification causes difficulty since this problem cannot be solved directly by any of the existing techniques other than enumeration.

In this model, the cycle time, T_s, given that the length of the queue is equal to n is given by

$$S_n = \sum_{\ell=1}^{n} X_\ell(q_n) , \tag{3.34}$$

where $\{X_\ell(q_n)\}_{\ell=1}^{n}$ is a sequence of independent but not identically distributed random variables. $X_\ell(q_n)$ is the ℓth quantum service, since the instant of the beginning of the current cycle, which has a distribution $F_{n_\ell}(x)$ given by (3.3), where n_ℓ is the queue length just before the service starts. The transition probability matrix, $P^* = (p^*_{ij})$, at instants of completion of quanta of service is given by

$$p^*_{ij} = p_{ij}(q_i), \qquad \text{for} \quad i.j = 0, 1, \ldots, N, \tag{3.35}$$

where $p_{ij}(q_i)$ is given by (3.15). Consequently the transition probability matrix $A^* = (a^*_{ij})$ at those instants when new cycles begin is given by

$$(3.36) \qquad a^*_{ij} = \begin{cases} p^{*(i)}_{ij}, & j \geq 2, \\ p^{*(i)}_{i0} + p^{*(i)}_{i1}, & j = 1, \end{cases}$$

where $p^{*(i)}_{ij}$ is the (i,j)th element in the matrix p^{*i}.

Since quantum services are not identically distributed we need to derive our cost function. Let r_{ij} be the cost associated with the transition from i to j where $i, j = 1, \ldots, N$, then

$$(3.37) \qquad r_{ij} = E(S_i - \theta_1)^2 = \mathrm{Var}\left(\sum_{\ell=1}^{i} X_\ell\right) + \left(\theta_1 - E\left(\sum_{\ell=1}^{i} X_\ell\right)\right)^2.$$

We find from (3.6), (3.7), (3.15), (3.35), (3.36) and the Markovian property the following expressions

$$(3.38) \qquad E(S_i) = E\left(\sum_{\ell=1}^{i} X_\ell\right)$$

$$= \sum_{\ell=1}^{i} \sum_{k=i-1}^{N} \frac{p^{*(\ell)}_{ik}\, p^{*(i-\ell)}_{kj}\, a_{1,k}}{a^*_{ij}},$$

and

$$\text{(3.39)} \qquad \text{Var}(S_i) = \text{Var}\left(\sum_{\ell=1}^{i} X_\ell\right)$$

$$= \sum_{\ell=1}^{i} \sum_{k=i-1}^{N} \left(\frac{p_{ik}^{*(\ell)} p_{kj}^{*(i-\ell)}}{a_{ij}^{*}}\right)^2 (a_{2,k} - a_{1,k}).$$

Thus c_i^*, the cost associated with the state i is given by

$$\text{(3.40)} \qquad c_i^* = \sum_{j=1}^{N} a_{ij}^* r_{ij}^* .$$

Although the optimal policy for this model exists since the class of all policies, R, is finite, we shall not be able to find such a policy by the techniques of stochastic programming mentioned previously. The difficulty is the fact that the selection of the control parameter, quantum size, for one state may affect the transition probabilities and the cost functions for other states. Consequently the first part of Howard's technique can be used to compare different given policies, but the second part cannot be used to choose a better policy. One way to pick a better policy than the one already used, however impractical it might be, is by enumerating among the m^N policies and selecting the one which minimizes

$$(3.41) \qquad \sum_{\ell=1}^{N} \sum_{i=1}^{N} a_{\ell i}^{*\{k(i)\}} (c_i^{*\{k(i)\}} + \sum_j a_{ij}^{*\{k(i)\}} v_0(j)),$$

where $v_0(j)$ are those obtained from the first part of Howard's technique. Further investigation might lead to a more practical way than enumeration in order to improve the chosen policy.

The main disadvantages in all the techniques discussed previously is that alternatives for the control parameter, the quantum size, are assigned particular values according to intuition and belief of the human designer and not determined by the technique itself. Thus, the optimal policy is only optimal within the class of all policies considered and hence a most definite form of suboptimization is introduced. To overcome this disadvantage, one needs to increase the number of alternatives for the quantum size at each state. However, this will increase the amount of computations which might lead to the practical infeasibility of the technique. A second disadvantage is that the techniques require Markovian assumptions for both arrival and service times. Besides that, optimization of our system might only be considered for particular requests in the system since the system is not in equilibrium at any instant. Further, the methods provide a numerical solution to the problem which may cause computational difficulties, especially for large size problems. Since a method which provides an analytic expression to allocate the quantum size may

overcome such difficulties, the optimum control system approach is investigated.

Optimum Control System Approach

The goal in this approach is to design a decision rule in such a way that the system satisfies some stability property while improving our cost function. Thus the problem is formulated as a mathematical optimization problem which is discussed via the definition of Lyaponov function and the "second method" of Lyaponov (Aoki, 1967; Kushner, 1967). Although further work needs to be done for a complete analysis of the system by such a method, the discussion suggests a simple technique to optimize this system.

The "second method" of Lyaponov is considered the most general approach in the theory of stability of deterministic dynamic systems. Recently the idea of discussing stability of stochastic systems appeared by suitably extending the deterministic Lyaponov theory. The "second method" is in fact an idea contained in the following reasoning: A dynamic system is stable, in the sense it returns to equilibrium after any perturbation, if and only if there exists a "Lyaponov function." A Lyaponov function is some scalar function $V(X)$ which describes the state of the system and has the properties:

(3.43) a) $V(X) > 0$ and $\frac{dV(X)}{dt} < 0$ for $X \neq X_e$

b) $V(X) = \frac{dV(X)}{dt} = 0$ when $X = X_e$,

where X is the state variable and X_e is the state of the system at equilibrium.

The model considered here assumes that the maximum number of active users is infinite. Arrival times of requests are assumed to be mutually independent and identically distributed random variables which are independent of service time, and are exponentially distributed with mean $1/\lambda$ seconds. With the exception of these two assumptions this model is the same as the ones previously discussed.

In order to apply the "second method" to this model we describe the system by the function

(3.44) $$V(E(n), q_n) = E(T_s - \theta_1)^2.$$

The state variable of this system is the expected value of the number of requests in the system. q_n is the control variable, the quantum size, which needs to be adjusted according to the state of the system in such a way that the cycle time of any request in the system approaches θ_1. In order to find the conditions for regulating the cycle time, we need to find the set of values that q_n can take in

order $V(E(n), q_n)$ to be a Lyaponov function. We are assuming in this discussion continuity in the state variable, $E(n)$. We also assume continuous observation of the system. Although these assumptions are not necessary for the application of the Lyaponov method (Kalaman and Bertram, 1960), we adopt them for the simplicity of the discussion. We have $V(E(n), q_n)$ satisfies the non-negative requirements of Lyaponov function. Again for simplicity we will denote $V(E(n), q_n)$ by V. If the quantum size which is the design parameter (the control) is chosen in such a way that the value $V' = \frac{dV}{dt}$ is always negative, and if there exists an equilibrium state $E_e(n)$ where $V = V' = 0$ then V is a Lyaponov function.

In order to obtain a suitable expression for V in terms of the state and control variables, a more convenient expression of V is found.

$$V = E(T_s - \theta_1)^2 = \text{Var}(\theta_1) + (\theta_1 - E(T_s))^2 . \tag{3.45}$$

To express the cycle time, T_s, in terms of the number of requests in the system, n, we have

$$E(T_s) = E(E(T_s | n)),$$

and

$$\text{Var}(T_s) = E(\text{Var}(T_s | n)) + \text{Var}(E(T_s | n)), \tag{3.46}$$

where n is a random variable. At the instant of time when a new cycle is about to start, n is exactly known, and therefore

$$E(T_s) = E(T_s \mid n),$$

and

$$\text{Var}\,(T_s) = \text{Var}\,(T_s \mid n). \tag{3.47}$$

Since n is a fixed value at instants of inspection, then $E(n) = n$. By (3.6) and (3.7)

$$V = n(a_{2,n} - a_{1,n}^2) + (\theta_1 - n a_{1,n})^2, \tag{3.48}$$

where

$$a_{1,n} = \frac{1-e^{-\mu q_n}}{\mu} + \tau, \tag{3.6}$$

and

$$a_{2,n} = \tau^2 + \frac{2}{\mu}(1-e^{-\mu q_n})(\frac{1}{\mu}+\tau) - \frac{2q_n e^{-\mu q_n}}{\mu}. \tag{3.7}$$

To find V', we have,

$$V' = \frac{dV}{dE(n)} \quad \frac{dE(n)}{dt}, \tag{3.49}$$

where

$$\frac{dV}{dE(n)} = \frac{dV}{dn} = (a_{2,n} - a_{1,n}^2) - 2a_{1,n}(\theta_1 - n a_{1,n}). \tag{3.50}$$

Before we find $\frac{dE(n)}{dt}$ we shall consider two cases. The first is the

case when the swap time, τ, is equal to zero and the second is for nonzero swap time.

By setting $\tau = 0$ in (3.6) and (3.7) and substituting in (3.50) we have

$$(3.51)\quad \frac{dV}{dE(n)} = \left(\frac{1}{\mu^2} - \frac{2\theta_1}{\mu} + \frac{2n}{\mu^2}\right) + e^{-\mu q_n}\left(\frac{2\theta_1}{\mu} - \frac{4n}{\mu^2} - \frac{2q_n}{\mu}\right) + e^{-\mu q_n}\left(\frac{2n-1}{\mu^2}\right).$$

To find $\frac{dE(n)}{dt}$, let n_t be the queue length at the start of a cycle, instant t. The length of the queue $n_{t'}$, at the moment when the next cycle starts, t', is given by

$$(3.52)\quad n_{t'} = n_t e^{-\mu q_{n_t}} + \lambda n_t E(X(q_{n_t})),$$

where $X(q_{n_t})$ is the quantum service of a request which is allocated a quantum size q_{n_t}. Consequently,

$$(3.53)\quad \frac{dE(n)}{dt} = \frac{n_{t'} - n_t}{n_t E(X(q_{n_t}))} = \lambda - \frac{1 - e^{-\mu q_{n_t}}}{E(X(q_{n_t}))} = \lambda - \mu .$$

This is the rate of change of the number of requests in the system as long as the system is not empty. When there are no requests in the system to be served, $\frac{dE(n)}{dt} = \lambda$. Hence at any time,

$$\frac{dE(n)}{dt} = \lambda - \mu + \delta_{n0}(t)\mu, \tag{3.54}$$

where

$$\delta_{n0} = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0. \end{cases} \tag{3.55}$$

Expression (3.54) is an exact form (Cox, 1961), since arrival and service times are exponential. It also shows that the number of requests in the system does not depend on the control, q_n. This is due to the fact that the limiting queue size distribution is independent of the queue discipline as long as the distribution of the service times and their independence is maintained (Takacs, 1962). Notice that if the maximum number of users is finite this will not be the case. Since $\frac{dE(n)}{dt}$ is independent of the control variable, the system is a free system and as $t \to \infty$ the system reaches equilibrium if $\lambda < \mu$. At equilibrium the expected number of requests in the system is (Cox, 1961),

$$E(n) = \frac{\lambda}{\mu - \lambda} . \tag{3.56}$$

In order to regulate the cycle time, T_s, at any instant one should select the value of q_n which minimizes the cost function, V, with respect to the number of requests in the system at that instant. This can be done by solving for q_n in

$$\frac{dV}{dn} = \frac{dV}{dE(n)} = A e^{-2\mu q_n} + B e^{-\mu q_n} \quad C = 0, \tag{3.57}$$

where (from (3.51))

$$A = \frac{2n-1}{\mu^2} \tag{3.58}$$

$$B = \frac{2\theta_1}{\mu} - \frac{4n}{\mu^2} - \frac{2q_n}{\mu},$$

$$C = \frac{1}{\mu^2} - \frac{2\theta_1}{\mu} + \frac{2n}{\mu^2}.$$

solutions to (3.57) can be found from

$$e^{-\mu q_n} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{3.59}$$

where

$$\left(\frac{d^2V}{dn^2}\right)_{q_n^{\min}} > 0. \tag{3.60}$$

($q_n^{\min}$ denotes the positive real solution to (3.59)). Note that B is a function of q_n and thus (3.59) must be solved iteratively and may yield multiple values for q_n. It is easily verified that $q_n = 0$ is one solution and that it is associated with

$$e^{-\mu q_n} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{3.61}$$

In order that the other solution to (3.59) yields a positive value of q_n, we must have

$$0 < \frac{-B - \sqrt{B^2 - 4AC}}{2A} < 1 . \tag{3.62}$$

This implies $B < 0$ since $A > 0$; that is,

$$\frac{\theta_1}{\mu} - \frac{2n}{\mu^2} - \frac{q_n}{\mu} < 0 , \tag{3.63}$$

thus

$$q_n > \theta_1 - \frac{2n}{\mu} .$$

Furthermore, for $-B - \sqrt{B^2 - 4AC} > 0$ we must have $4AC > 0$, thus $C > 0$, or

$$\frac{1}{\mu^2} - \frac{2\theta_1}{\mu} + \frac{2n}{\mu^2} > 0 . \tag{3.64}$$

We then have the following restriction on the desired mean cycle time

$$\theta_1 < \frac{2n+1}{2\mu} . \tag{3.65}$$

Since $n \geq 1$, $\theta_1 < \frac{3}{2\mu}$ is a necessary condition for $q_n^{\min}$ to exist.

Thus for $\tau = 0$ one can find the decision procedure $\{q_n\}$, for which the cost function V is minimized at any instant. Such decision procedure will not affect in any way the future length of the queue.

To discuss the system for nonzero swap time, one can find from a similar argument as above that the rate of change of the

number of request in the system when the system is not empty is

$$(3.66) \qquad \frac{dn}{dt} = \frac{dE(n)}{dt} = \frac{n'-n}{E(X(q_n))} = \lambda - \frac{\mu(1-e^{-\mu q_n})}{(1-e^{-\mu q_n})+\tau\mu} .$$

Here, a choice of a quantum size q_n affects the future number of requests in the system. It is clear that the larger q_n, the shorter the length of the queue will be since we incur less swap time. Although we adopt the following argument which depends on our knowledge about the system, we will show graphically in the Appendix that the same results can be found by applying the "second method." From (3.61), one can view this system as if requests need a prolonged service time with mean, $\frac{1}{\mu'}$,

$$(3.67) \qquad \frac{1}{\mu'} = \frac{1}{\mu} + \frac{\tau}{1-e^{-\mu q_n}} ,$$

i.e., this service is approximately exponential. Again for a request which needs service X, the smaller the quantum size, the longer this request's actual service time because of additional swap time. The assumption that the distribution of the actual service time X', $f_{X'}(x')$, is approximately exponential may not weaken the results significantly (Greenberger, 1966) and hence (3.61) is a reasonable approximation (Cox, 1961). If the rate of arrival is greater than the service rate of a request, then the queue will tend towards infinity.

Thus, in order to achieve equilibrium we must have,

$$\lambda < \frac{\mu(1-e^{-\mu q_n})}{(1-e^{-\mu q_n})+\tau\mu} , \tag{3.68}$$

and consequently the minimum quantum size which assures stability, given that there are n requests in the system, q^{stab}, is

$$q^{stab} = -\frac{1}{\mu} \ln \left(\frac{\lambda-\mu+\lambda\tau\mu}{\lambda-\mu}\right) + \epsilon \tag{3.69}$$

where $\epsilon > 0$. As $t \to \infty$ the system will reach stability if $q_n \geq q^{stab}$. Note that q^{stab} is negative if $\lambda > \mu$.

On the other hand the value of q_n which minimizes V at any instant can be found by substituting (3.6) and (3.7) in (3.50) and solving for q_n in,

$$\frac{dV}{dn} = \frac{dV}{dE(n)} = A e^{-2\mu q_n} + B e^{-\mu q_n} + C = 0 \tag{3.70}$$

where

$$A = \frac{2n-1}{\mu^2} , \tag{3.71}$$

$$B = \frac{2\theta_1}{\mu} - \frac{4n}{\mu^2} - \frac{2q_n}{\mu} - \frac{4n\tau}{\mu} ,$$

$$C = \frac{1}{\mu^2} + \frac{2n}{\mu^2} - \frac{2\theta_1}{\mu} + \frac{4n\tau}{\mu} + 2n\tau^2 - 2\tau\theta_1 .$$

and similar to the above argument one can find a necessary condition for q_n^{min} to exist, for all values of n, is

$$\theta_1 < \frac{1}{(1+\mu\tau)} \left(\frac{3}{2\mu} + 2\tau + \mu\tau^2\right). \tag{3.72}$$

Since a choice of q_n at any time will affect the future length of the queue one needs to choose q_n from the set of values that satisfies stability of the system while keeping the cycle as close as possible to the desired cycle time θ_1. Thus one chooses for any n, $n \geq 1$ a quantum size that is equal

$$q_n^{opt} = \max (q^{stab}, q_n^{min}) . \tag{3.73}$$

Rather than calculate all possible solutions for q_n^{min} from an expression corresponding to that in (3.59) we present in the Appendix graphs of V' with respect to a wide range of parameter values. One can see from these graphs that the value of q_n^{opt} and the right most root of V' coincide. The Appendix also illustrates that the function V' is always negative for all values of $q > q_n^{opt}$, where V' is

$$V' = [e^{-2\mu q_n}\left(\frac{2n-1}{\mu^2}\right) + e^{-\mu q_n}\left(\frac{2\theta_1}{\mu} - \frac{4n}{\mu^2} - \frac{2q_n}{\mu} - \frac{4n\tau}{\mu}\right)$$

$$+ \left(\frac{1}{\mu^2} + \frac{2n}{\mu^2} - \frac{2\theta_1}{\mu} + \frac{4n\tau}{\mu} + 2n\tau^2 - 2\tau\theta_1\right)] \tag{3.74}$$

$$\times \left[\lambda - \frac{\mu(1-e^{-\mu q_n})}{(1-e^{-\mu q_n})+\tau\mu}\right].$$

If there exists a state n such that $V(n, q_n^{opt}) = 0$ and hence $V'(n, q_n^{opt}) = 0$ then V is a Lyapunov function and the state n is the equilibrium state.

Thus by means of the above method we choose the quantum size, at a given state, such that it satisfied some stability property and also minimizes V for this state. Such method gives a heuristic solution to our problem when other methods may not be computationally feasible. One should notice that this method is applicable without any distribution assumptions on any of the variable involved. Further applications of this method on other time sharing models seems to be practical. Chapter IV will give numerical examples of the techniques described in this Chapter.

IV. COMPUTATIONAL RESULTS AND CONCLUSIONS

Numerical Examples

In the last chapter we presented two techniques by which one can find a decision procedure which allows the quantum to vary with the number of users in the system, in order to meet certain operational requirements. We now discuss the computational aspects of each technique.[3]

The first technique for finding the optimal policy (decision procedure) which allocates the quantum dynamically for a finite number of states and alternatives is based upon Howard's algorithm. The technique was tested on the case where the maximum number of users was five, $(N = 5)$ and the number of alternatives for quantum sizes was five $(k = 5)$. The alternatives for the quantum in each state were .1, .2, .3, .4 and .5 seconds. The rate of arrival λ and the rate of service μ were equal to .5/second and .1/second respectively. Table 1 gives the optimum quantum size in seconds for different values of, swap time, τ, and desired cycle time, θ_1, as calculated by this technique.

[3]All computations were done on the CDC 3300 at Oregon State University.

Table 4.1. Values of q^{opt} for $\lambda = .5$ and $\mu = 1$.

State	$\theta_1 = .5$			$\theta_1 = 1.0$			$\theta_1 = 1.5$		
No.	$\tau = .01$	$\tau = .05$	$\tau = .1$	$\tau = .01$	$\tau = .05$	$\tau = .1$	$\tau = .01$	$\tau = .05$	$\tau = .1$
1	.5	.5	.5	.5	.5	.5	.5	.5	.5
2	.3	.2	.2	.5	.4	.4	.5	.5	.5
3	.2	.1	.1	.3	.3	.2	.5	.4	.3
4	.1	.1	.1	.2	.2	.1	.3	.3	.2
5	.1	.1	.1	.2	.1	.1	.3	.2	.1

The results demonstrate the intuitive idea that the more we increase the desired cycle time, θ_1, the more we can increase the quantum allocated to serve each request without degrading the system. Correspondingly, as the swap time increases, the quantum size would decrease. In particular, as the swap time increases for a fixed desired cycle time the quantum may decrease to a point where the queue is saturated. For example, when $\theta_1 = .5$ and $\tau = .1$ then once state 3, 4 or 5 is reached the computer will only be swapping and be doing no processing of the user's request. This indicates that the desired cycle time is too close to the amount of swap time required for certain states.

The program took five seconds of computer time to give each of the solutions in the above table. When it was run for the case $N = 12$ and $k = 12$, it took approximately 222 seconds of computer time to compute three iterations and the method had not yet converged. This indicates the computational costs involved in a system where N

and k are large. Also, in our applications of the technique we found situations where some of the coefficients of the transition probability matrix, A, were zero when calculated by means of expressions (3.15) and (3.16). Since the analytic results are under the assumption that the chain is completely ergodic (which implies the requirement of strictly positive elements of the matrix A), we must fill in the zeros in A before we can apply the technique. One way to do this is to multiply the matrix by itself a few times until all zeros disappear (this will not affect the solution since we are seeking the limiting distribution). One can easily see that this can be done in a finite number of times since the chain is ergodic. Another way of eliminating zeros is to replace the zero elements with small quantities. This approximation might not have serious effect on the result, especially when it is compared with the round off error incurred in calculating the elements of the matrix A. Such round off error is anticipated to affect the results, in particular when N is large. Due to the fact that the computational requirements to solve a medium size problem are excessive, it is impractical at this point to demonstrate the results for a large hypothetical system.

The second technique provides means by which one can find the optimal quantum, q_n^{opt}, when the number of active users in the system is n. A number of graphs were obtained to study the changes in q_n^{opt} corresponding to changes in the state of the system.

Figures 4.1, 4.2 and 4.3 show the optimum quantum vs. the number of active users, n, for different swap times (note that in Figure 4.1, $\tau = 0$ and therefore λ is not involved). It can be noted that as μ decreases q_n^{opt} decreases. One can also notice the increase in q_n^{opt} corresponding to an increase in the value of θ_1. Further, when the number of active users in the system, n, is small, the increase in q_n^{opt} is significant with changes in the number of users; however, as n becomes large (above a particular value) the change in q_n^{opt} is small with the changes in the number of active users. The effect of τ on q_n^{opt} can be seen by comparing the three figures. Specifically, as τ increases q_n^{opt} decreases.

In order to study the interaction of q_n^{opt} for a dynamic system, preliminary simulation runs were obtained. The simulated model deviated from the theoretical one in that the number of users was assumed finite. The results showed significant improvements in regulating the cycle time. In order to recognize such improvements, simulation results of this model were compared with the simple round robin, RR, which allocates the same quantum to all states. To distinguish between the two models in the following discussion we will denote our model as round robin with dynamic quantum, RRDQ. In the two simulated systems, RR and RRDQ, requests were assumed to arrive according to the Poisson distribution with

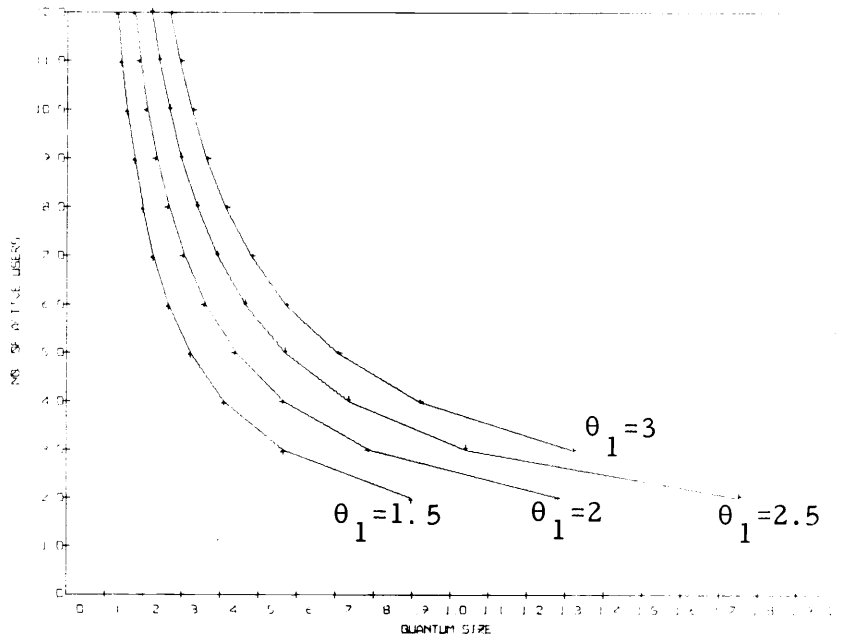

Figure 4.1.a. $\mu = .5$.

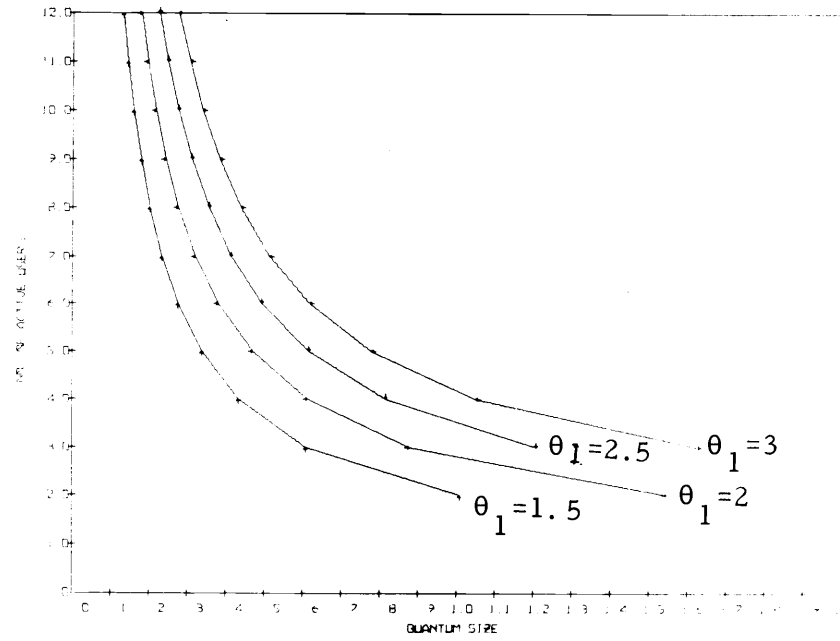

Figure 4.1.b. $\mu = .75$

Figure 4.1. The optimal quantum vs. the active number of users in the system, $\tau = 0$.

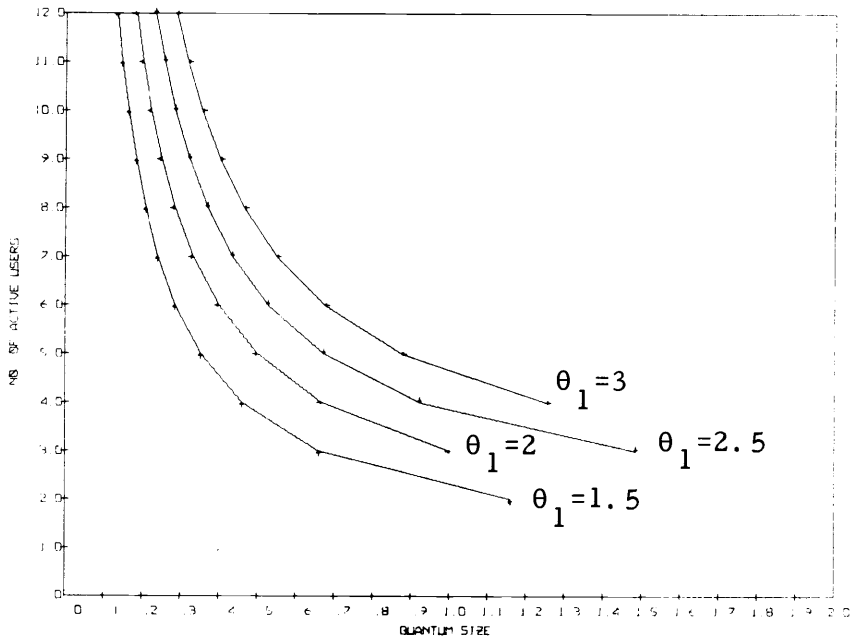

Figure 4.1.c. $\mu = 1$

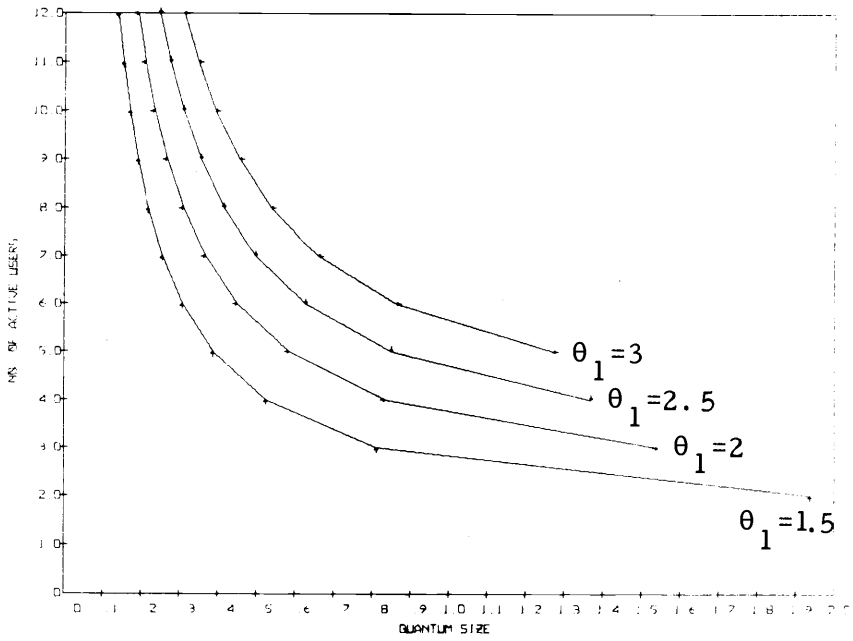

Figure 4.1.d. $\mu = 1.5$.

Figure 4.1. (continued)

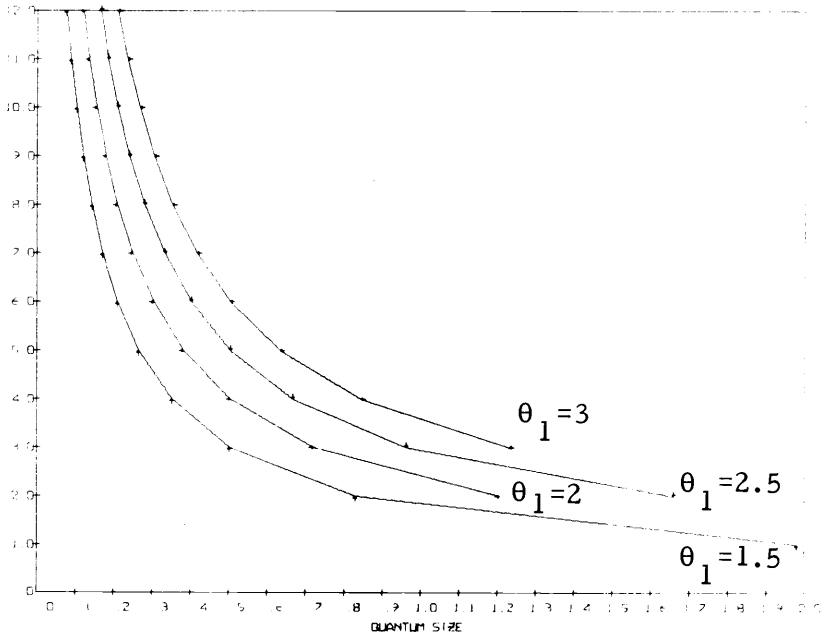

Figure 4.2.a. $\mu = .5$

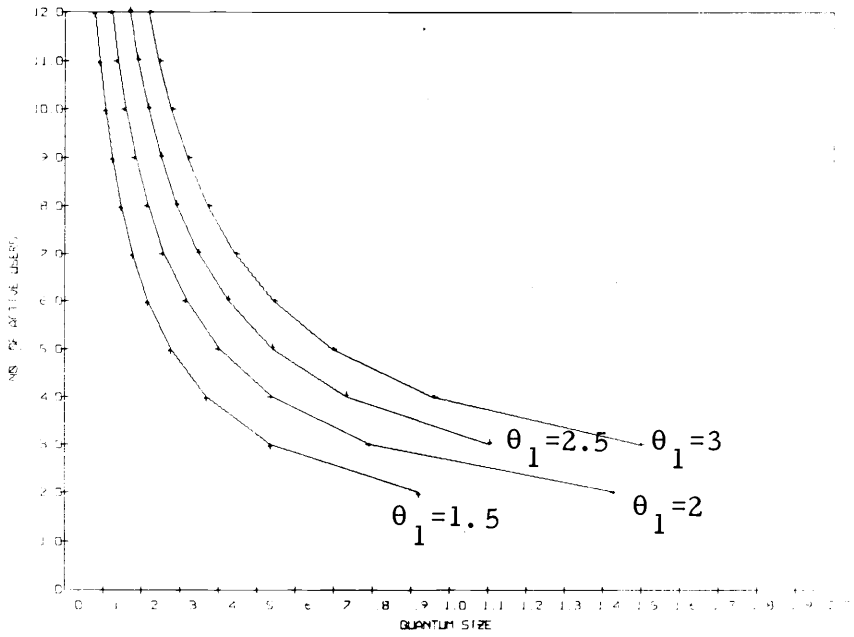

Figure 4.2.b. $\mu = .75$.

Figure 4.2. Optimal quantum vs. the number of active users in the system, $\tau = .05$, $\lambda = .3$.

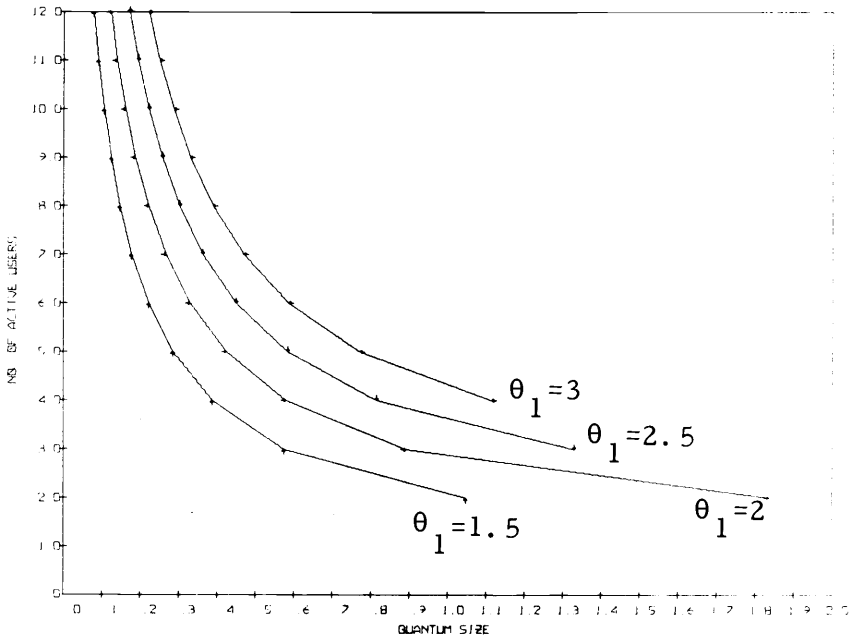

Figure 4.2.c. $\mu = 1.0$.

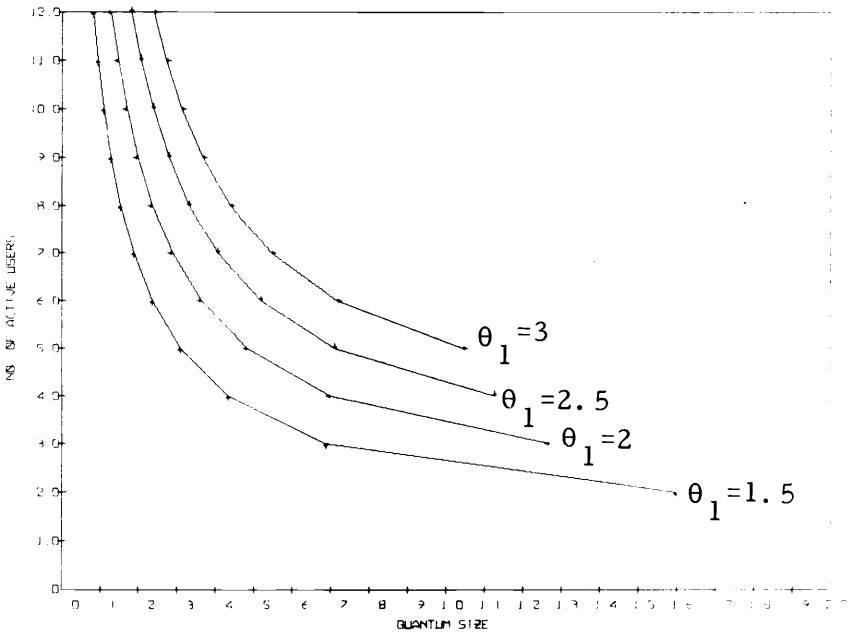

Figure 4.2.d. $\mu = 1.5$.

Figure 4.2. (continued)

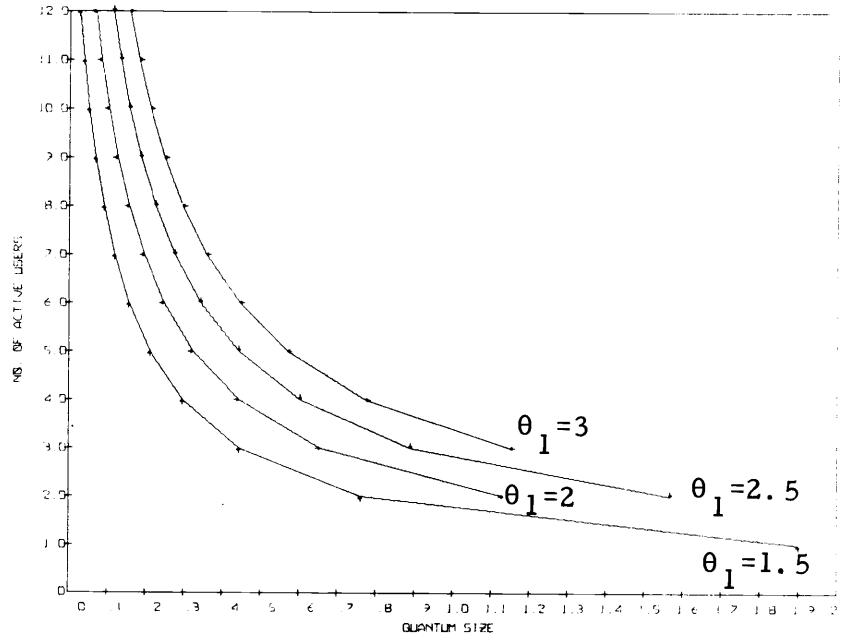

Figure 4.3.a. $\mu = .5$.

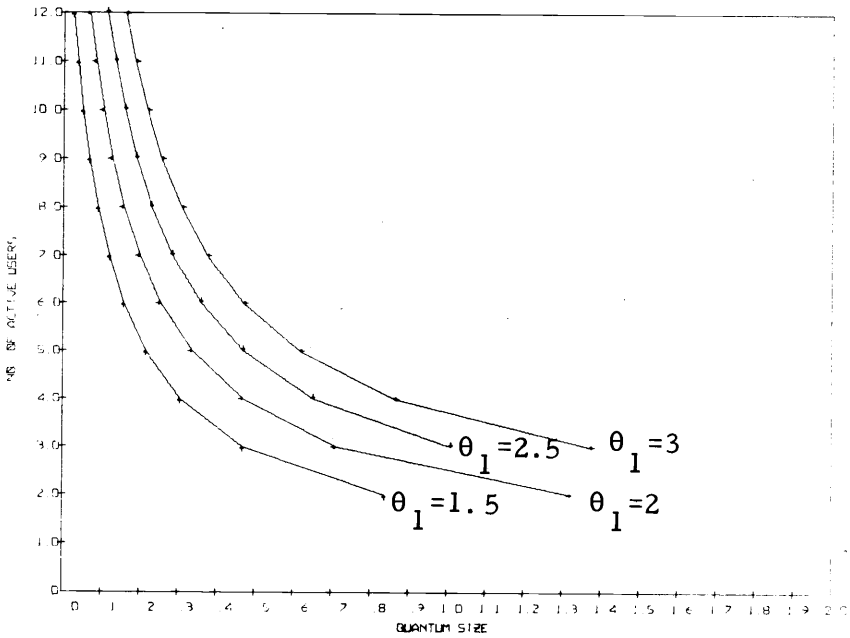

Figure 4.3.b. $\mu = .75$.

Figure 4.3. Optimal quantum vs. the number of active users in the system, $\tau = .1$, $\lambda = .3$.

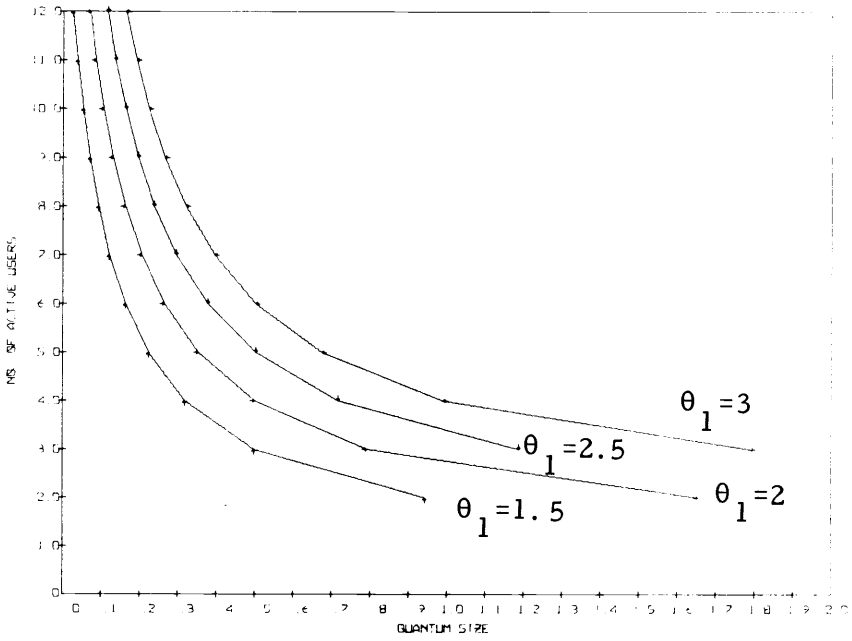

Figure 4.3.c. $\mu = 1$.

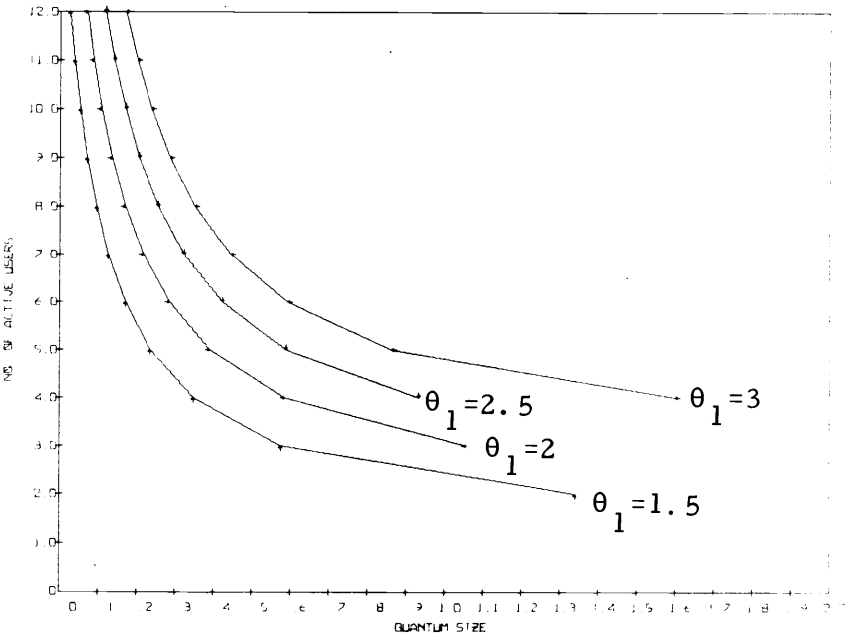

Figure 4.3.d. $\mu = 1.5$.

Figure 4.3. (continued)

rate λ per second. It was assumed that $N = 20$, thus if the number of active users in the system was less than 20 when a new request arrived, it was allowed to enter the queue of active users, otherwise it was lost. The service time of requests was assumed to be exponential with mean $1/\mu$ seconds and the swap time, τ, was taken to be zero. The cycle time was measured for each request, (not only the tagged requests) each time it entered the queue. The first cycle time was considered as the time between the request's arrival and its departure of the service facility (either to arrive again or leave the system). For the RRDQ, observation of the system and allocation of the optimal quantum was done after each quantum service. Table 4.2 illustrates simulation results of the two models. In order to compare the two models the allocated quantum, $\bar{q}$, in the RR system was chosen to be the average quantum size in the RRDQ model. The quantum distribution for RRDQ models is illustrated in Table 4.3. From Table 4.2, it can be seen that the variance was more reduced in the RRDQ system than in the RR system. Moreover, the mean and the variance of the requests waiting time in the system were almost the same in both models. Also the total number of requests processed during the eight hours of simulation were approximately the same. This is intuitively clear since $\tau = 0$. Although there are differences between the RR and RRDQ for these simulations we do not have enough runs to make any statistical

Table 4.2. Simulation results for both the RR and RRDQ models for $\lambda = .6$, $\mu = .8$ and $\tau = 0$.

Estimated variables	$\theta_1 = 1$		$\theta_1 = 1.5$		$\theta_1 = 2$	
	RR	RRDQ	RR	RRDQ	RR	RRDQ
Average quantum size $\bar{q}$	.22		.36		.55	
Mean cycle time	.8	.9	1.4	1.4	2.1	1.8
Variance of the cycle time	.57	.18	1.74	.32	3.33	.48
Mean waiting time	5.5	5.4	4.5	4.5	4.5	4.5
Variance of waiting time	56.7	57.6	49.8	53.1	47.4	52.2
Number of jobs processed during the 8 hours of simulation	17264	17266	17296	17296	17296	17296

Table 4.3. The optimum quantum and estimated distribution of the queue length.

n	$\theta_1 = 1$		$\theta_1 = 1.5$		$\theta_1 = 2$	
	q^{opt}_{n+1}	Queue length	q^{opt}_{n+1}	Queue length	q^{opt}_{n+1}	Queue length
1	.61	.118	1.03	.129	1.62	.153
2	.38	.097	.62	.089	.90	.084
3	.28	.101	.44	.093	.62	.093
4	.22	.100	.34	.096	.47	.097
5	.18	.090	.28	.091	.38	.090
6	.15	.080	.23	.083	.32	.080
7	.13	.070	.20	.068	.28	.069
8	.12	.063	.18	.058	.24	.055
9	.10	.051	.16	.047	.22	.046
10	.09	.042	.14	.038	.20	.040
11	.09	.030	.13	.029	.18	.030
12	.08	.028	.12	.025	.16	.027
13	.07	.026	.11	.024	.15	.023
14	.07	.020	.10	.016	.14	.020
15	.06	.016	.10	.014	.13	.020
16	.06	.014	.09	.013	.12	.018
17	.06	.018	.09	.011	.12	.019
18	.05	.015	.08	.012	.11	.015
19	.05	.012	.08	.013	.10	.013
20	.05	.009	.07	.054	.10	.010

statement at this point. Simulation study for the case $\tau > 0$ would be interesting as one can measure the efficiency of the processor by comparing the number of requests processed during a certain period by means of the RRDQ to those of the FCFS, which is the most efficient model.

Conclusions

In this thesis we have investigated a class of time sharing system models that allow for dynamic allocation of quantum size. In general such models are more flexible than the normal ones found in the literature. The major advantage of such models is their capability of reducing swap and overhead times while maintaining tolerable waiting times. This is accomplished by varying the quantum size with the state of the system while requiring the system to respond in a satisfactory way to the common user. The cost function to be minimized is the mean square difference between the cycle time and the desired cycle time. This cost function is a different criterion than the conventional ones used when analyzing mathematical time sharing models. It is based on the idea that there is no reason to believe that the purpose of time sharing is to minimize the response time for the small users whatever loss is incurred in swapping and overhead operations. Use of such criteria may not be of any value to the privileged user himself, since he has to spend a considerable

amount of time thinking before submitting his next request. To judge the system by our cost function seems more sensible than the one which favors short requests. The trade off between efficient use of the processor and favoring short requests is done dynamically dependent on the state of the system.

In order to investigate this class of systems, we discussed two techniques: the first dealing with a finite number of users and the second dealing with an infinite number of users. Although the first technique showed computational infeasibility in some cases, the second one, which is based upon a simple concept, provides a scheduling algorithm which shows improvement in system performance, when it is compared to the standard ones.

The two methods point out several aspects which should be considered in further research. In both techniques we are regulating the cycle time only for tagged requests. The behavior of the cycle time for all requests should be investigated. One possibility using the stochastic programming approach might be to formulate a semi-Markov model and optimize it by a method suggested by Jewell (1963). Dynamic quantum allocation should also be investigated for different types of cost functions and more general models. The optimal control methods need further development to allow for a finite number of users and possibly "better" Lyapunov functions than the ones presented. Further simulation investigation is also needed and may

provide a statistical measure of the efficiency of various dynamic models.

We believe that this thesis provides the background for the development of these areas of research.

BIBLIOGRAPHY

Aoki, M. 1967. Optimization of stochastic systems. New York, Academic. 354 p.

Chang, W. 1966. A queuing model for a simple case of time sharing. IBM System Journal 5:115-125.

Blackwell, D. 1961. On the functional equation of dynamic programming. Journal of the Mathematical Analysis and Applications 2:273-276.

Cobham, A. 1954. Priority assignment in waiting line problems. Operations Research 2:70-76.

Coffman, E.G., Jr. 1968. Analysis of two time-sharing algorithms designed for limited swapping. Journal of the Association for Computing Machinery 15:341-353.

Coffman, E.G., Jr. and L. Kleinrock. 1968a. Feedback queueing models for time-shared systems. Journal of the Association for Computing Machinery 15:549-576.

_____ 1968b. Computer Scheduling Methods and their Countermeasures. In: Proceedings of the Spring Joint Computer Conference of the American Federation of Information Processing Societies. Vol. 32 [Santa Monica, California] p. 11-21.

Corbato, F.J. et al. 1963. The compatable time-sharing system. Cambridge, MIT. 96 p.

Cox, D.R. and W.L. Smith. 1961. Queues. New York, Wiley. 180 p.

Derman, C. 1962. On sequential decisions and Markov chains. Management Science 9:16-24.

Estrin, G. and L. Kleinrock. 1967. Measures, models and measurements for time-shared computer utilities. In: Proceedings of 22nd National Conference of the Association for Computing Machinery. [New York] p. 85-96. (A.C.M. Publication P-67)

Fine, G.H. and R.V. McIsaac. 1966. Simulation of a time sharing system. Management Science 12:180-194.

Greenberger, M. 1966. The priority problem and computer time sharing. Management Science 12:888-906.

Hellerman, H. 1969. Some principles of time sharing scheduler strategies. IBM System Journal 2:94-117.

Howard, R.A. 1960. Dynamic programming and Markov processes. New York, Wiley. 136 p.

Jewell, W.S. 1963. Markov-renewal programming. Operations Research 11:938-971.

Kalman, R.E. and J.E. Bertram. 1960. Control system analysis and design via the "second method" of Lyapunov. Transactions of the ASME 82:371-400.

Kleinrock, L. 1964. Analysis of a time-shared system. Proceedings of the Naval Research Logistics Quarterly 11:59-73.

_____ 1967. Time-shared systems: a theoretical treatment. Journal of the Association for Computing Machinery 14:242-261.

Krishnamoorthi, B. and R.C. Wood. 1966. Time shared computer operations with both interarrival and service times exponential. Journal of the Association for Computing Machinery 13:317-338.

Kushner, H.J. 1967. Stochastic stability and control. New York, Academic. 161 p.

McCarthy, J.S., E. Boilen and J.C.R. Licklider. 1963. A time-sharing debugging system for a small computer. In: Proceedings of the Spring Joint Computer Conference of the American Federation of Information Processing Societies. Vol. 23. [Santa Monica, California] p. 51-57.

Manne, A. 1960. Linear programming and seqential decisions. Management Science 6:259-267.

Nielsen, N.R. 1967. The simulation of time sharing systems. Communication of the ACM 10:397-413.

O'Connor, T.J. 1965. Analysis of computer time-sharing system-a simulation study. Ph.D. thesis. Stanford, Stanford University 147 numb. leaves.

Phipps, T. E. 1956. Machine repair as a priority waiting-line problem. Operations Research 4:76-86.

Rasch, P. J. 1970. A queueing theory study of round robin scheduling of time-shared computer systems. Journal of the Association for Computing Machinery 17:131-145.

Saaty, T. L. 1961. Elements of queueing theory. New York, McGraw-Hill. 423 p.

Scherr, A. L. 1967. An analysis of time-shared computer systems. Cambridge, MIT. 115 p.

Schrage, L. E. 1967. The queue M/G/1 with feedback to lower priority queues. Management Science 13:466-474.

Schrage, L. E. and L. Miller. 1965. The queue M/G/1 with the shortest remaining processing time discipline. Operations Research 13:670-684.

Seaman, P. H. 1966. On teleprocessing system design. IBM System Journal 5:175-189.

Shapiro, S. 1965. Control waiting time in a queue. IBM System Journal 4:53-57.

Takacs, L. 1962. Introduction to the theory of queues. New York, Oxford University. 268 p.

White, D. J. 1969. Dynamic programming. San Francisco, Holden-Day. 180 p.

APPENDIX

APPENDIX

The graphs in this appendix are a plot of the V' function, given by (3.70), against the non-negative values of the quantum size. Any $q_n \geq q_n^r$, where q_n^r represents the right most root of V', will satisfy the Lyapunov condition $V' < 0$ for the range of parameters presented. Table A.1 gives the values of q_n^{opt} as calculated by (3.64), (3.66) and (3.69). It is easily seen that for the range of parameters plotted that q_n^{opt} is approximately equal to q_n^r, thus supporting the conjecture on page 70 that V is Lyapunov for the range $q_n \geq q_n^{opt}$.

Table A.1. Values of q_n^{opt} corresponding to parameter of Figure A.1.a. through A.1.j.

n	A.1.a. q_n^{opt}	A.1.b. q_n^{opt}	A.1.c. q_n^{opt}	A.1.d. q_n^{opt}	A.1.e. q_n^{opt}	A.1.f. q_n^{opt}	A.1.g. q_n^{opt}	A.1.h. q_n^{opt}	A.1.i. q_n^{opt}	A.1.j. q_n^{opt}
2	.565	.490	1.836	1.655	.828	.671	> 2	> 2	> 2	> 2
4	.221	.162	.581	.500	.250	.176	.938	.725	> 2	> 2
6	.124	.069	.330	.264	.132	.071	.406	.309	.991	.748
8	.078	.025	.222	.162	.081	.671	.253	.177	.509	.392
10	.051	0	.162	.105	.053	0	.177	.111	.342	.254
12	.034	0	.124	.069	.034	0	.132	.072	.254	.178

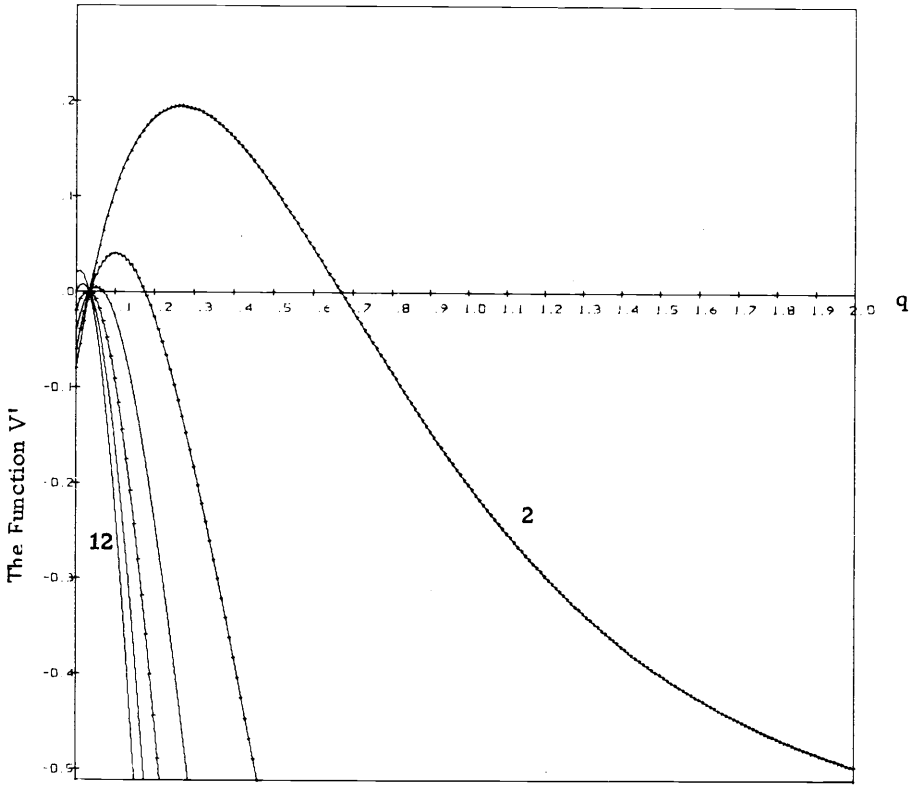

Figure A.1.a. $\lambda = .5$, $\mu = 1$, $\theta_1 = 1$, $\tau = .05$.

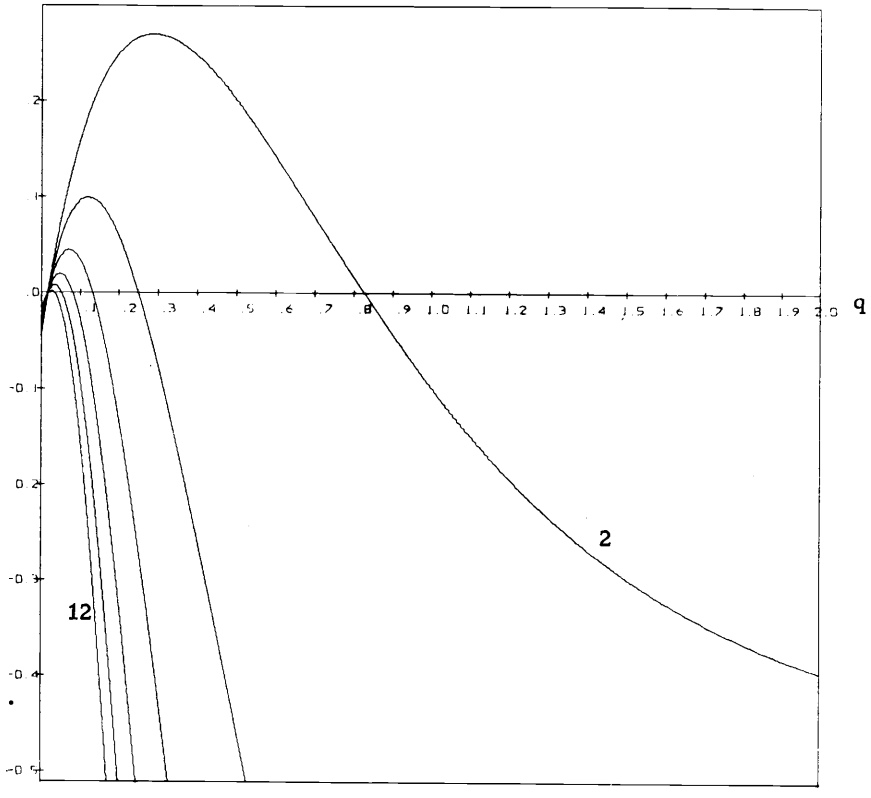

Figure A.1.b. $\lambda = .5$, $\mu = 1$, $\theta_1 = 1$, $\tau = .10$.

Figure A.1. The function V' vs. the quantum size, q.

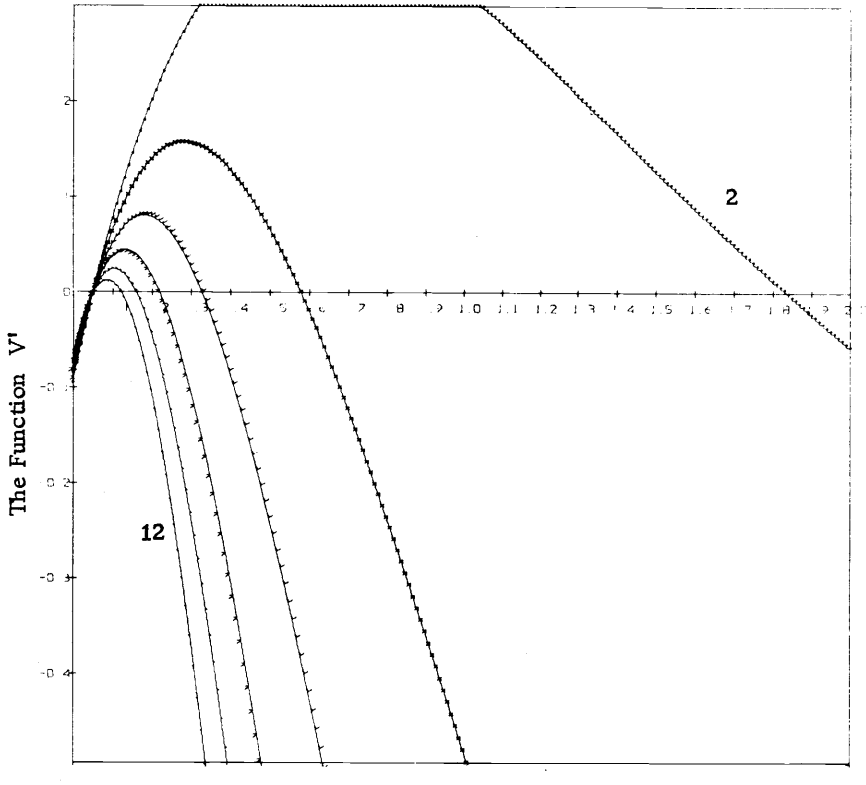

Figure A.1.c. $\lambda = .5$, $\mu = 1$, $\theta_1 = 2$, $\tau = .05$.

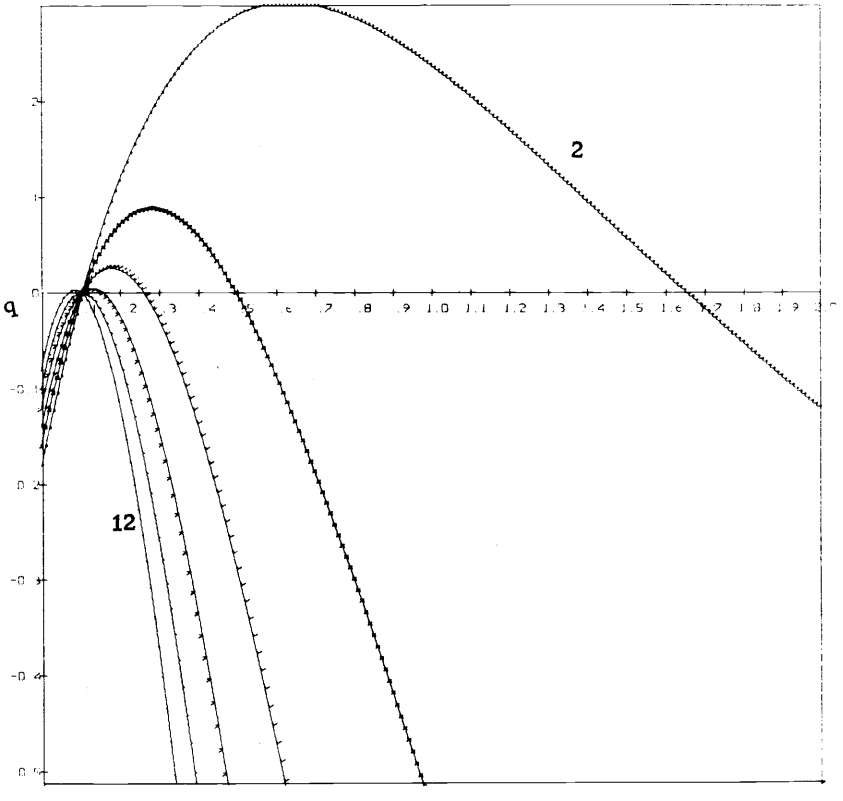

Figure A.1.d. $\lambda = .5$, $\mu = 1$, $\theta_1 = 2$, $\tau = .10$.

Figure A.1. (continued)

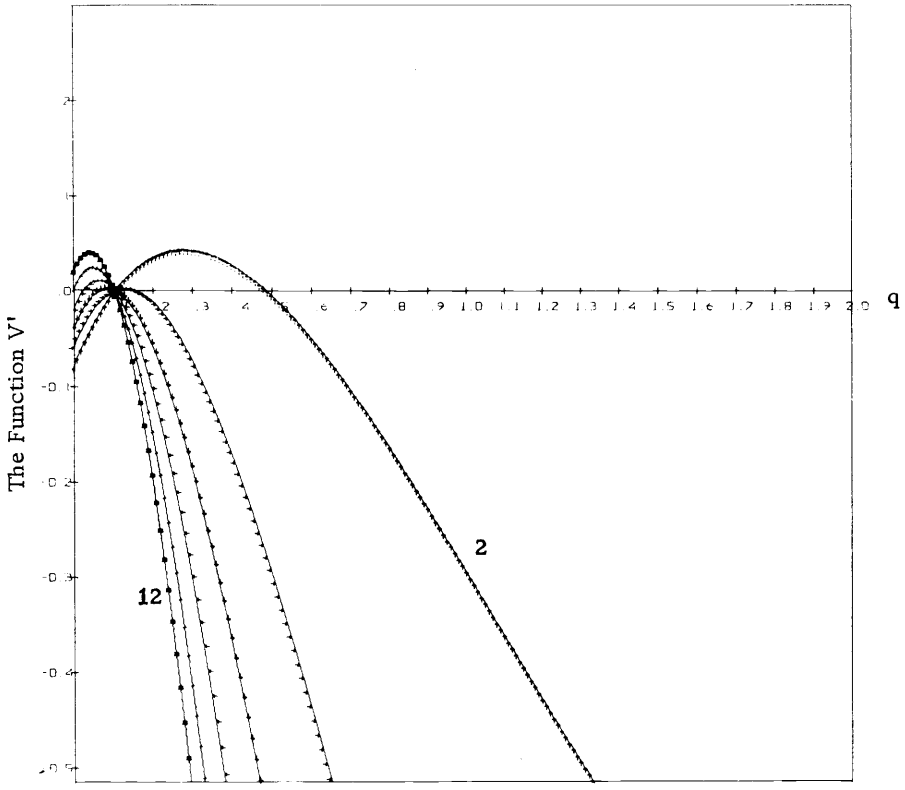

Figure A.1.e. $\lambda = .5$, $\mu = 2$, $\theta_1 = 1$, $\tau = .05$.

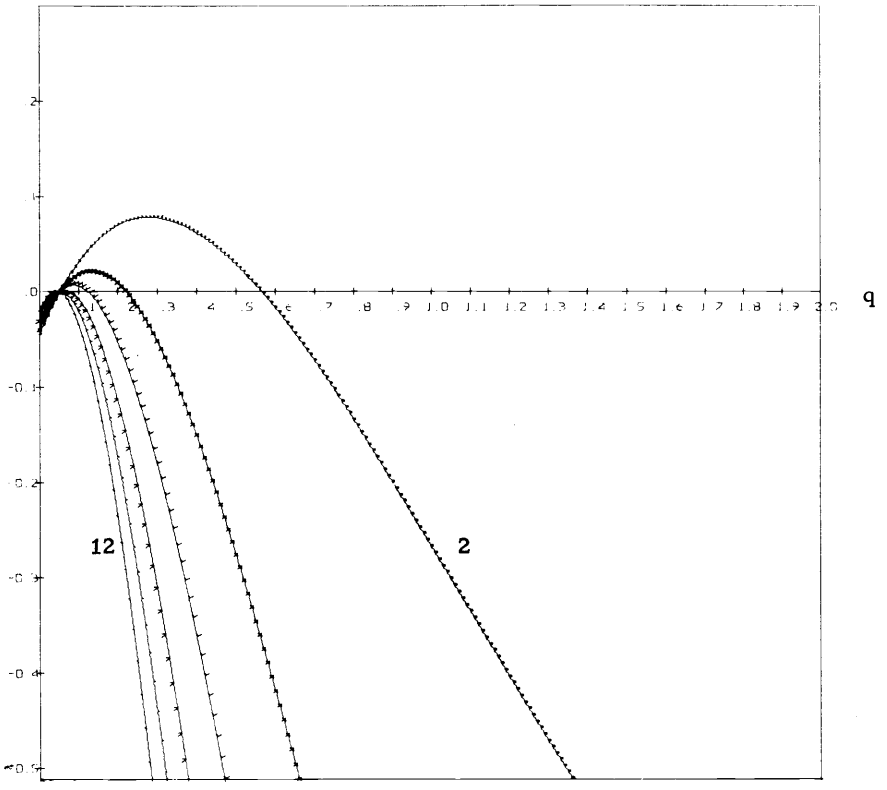

Figure A.1.f. $\lambda = .5$, $\mu = 2$, $\theta_1 = 1$, $\tau = .10$.

Figure A.1. (continued)

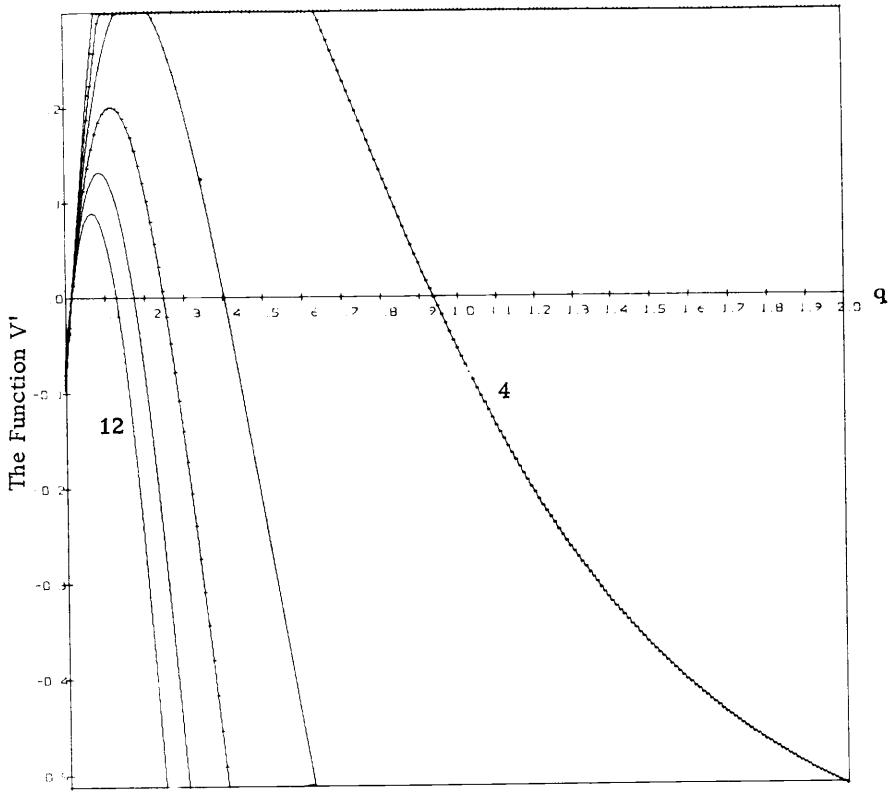

Figure A.1.g. $\lambda = .5$, $\mu = 2$, $\theta_1 = 2$, $\tau = .05$.

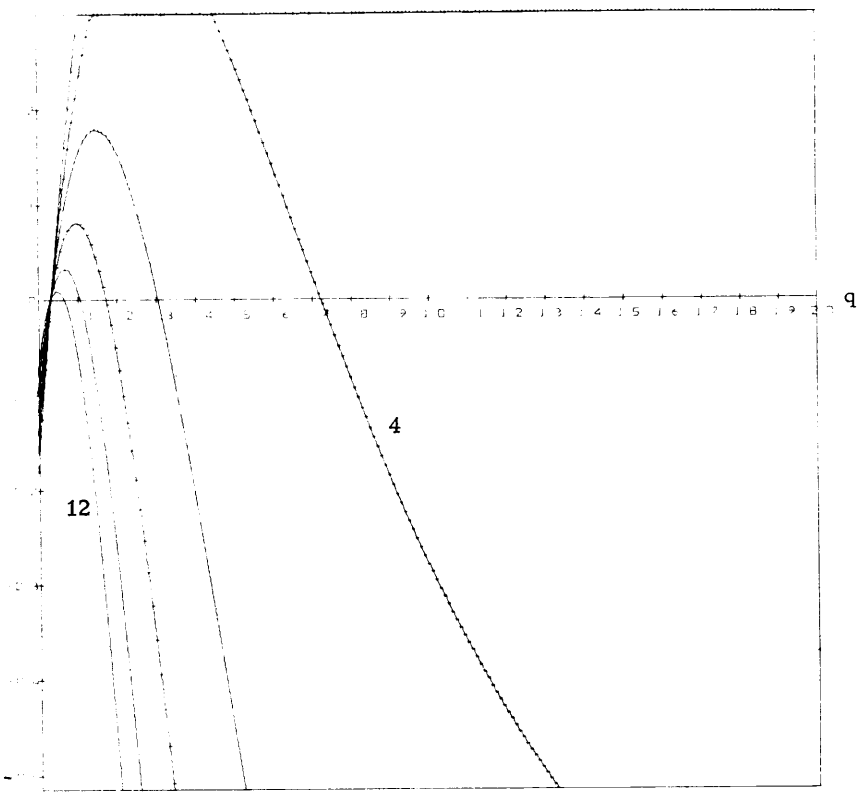

Figure A.1.h. $\lambda = .5$, $\mu = 2$, $\theta_1 = 2$, $\tau = .10$.

Figure A.1. (continued)

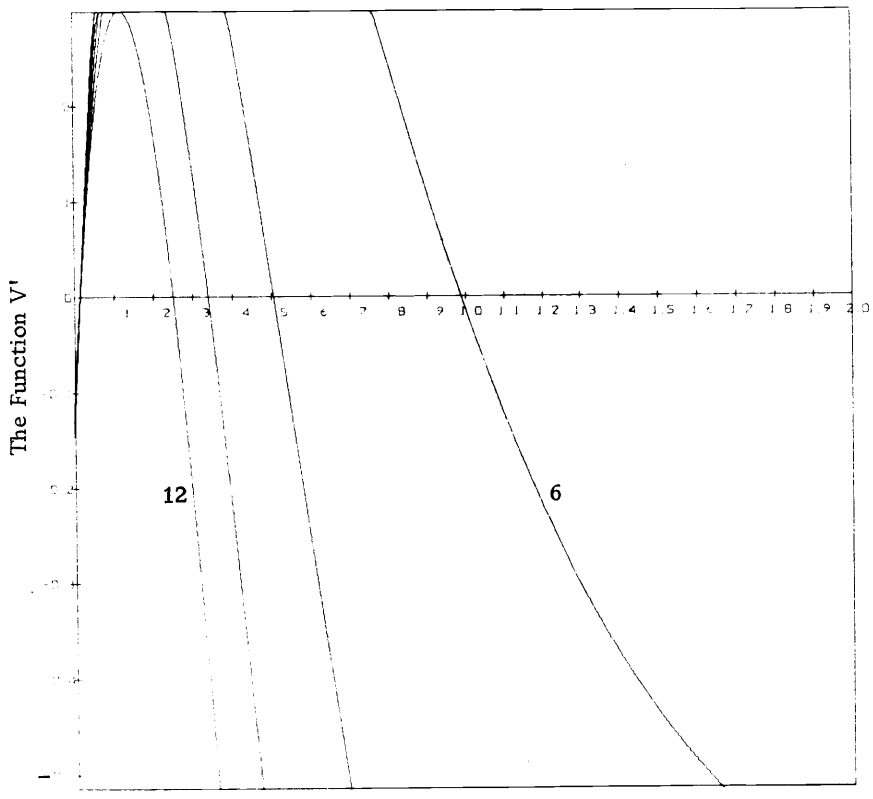

Figure A.1.i. $\lambda = .5$, $\mu = 2$, $\theta_1 = 3$, $\tau = .05$.

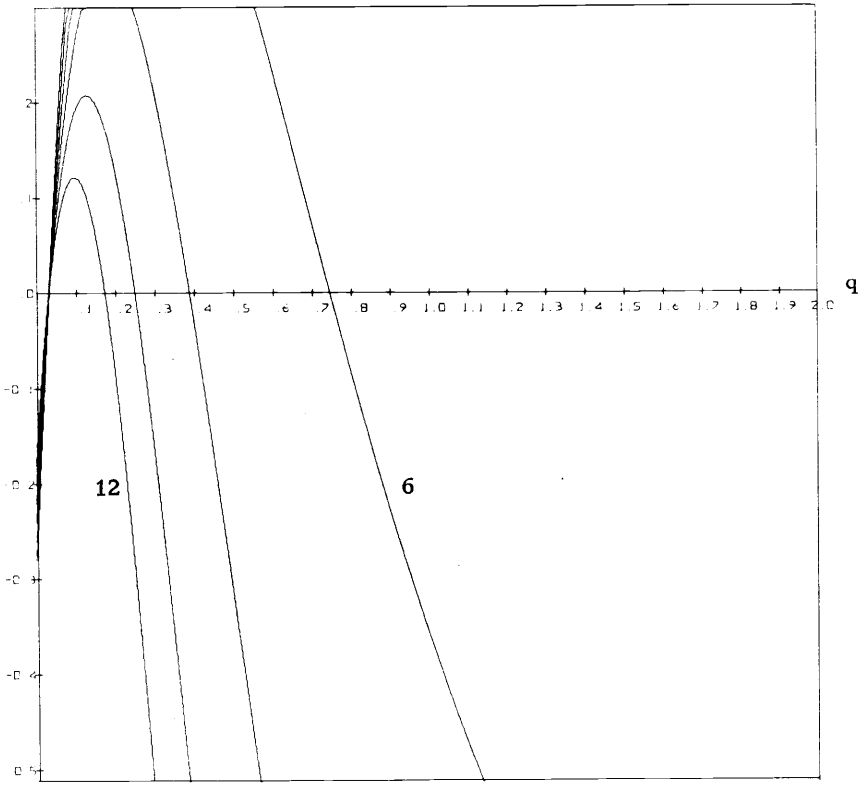

Figure A.1.j. $\lambda = .5$, $\mu = 2$, $\theta_1 = 3$, $\tau = .10$.

Figure A.1. (continued)

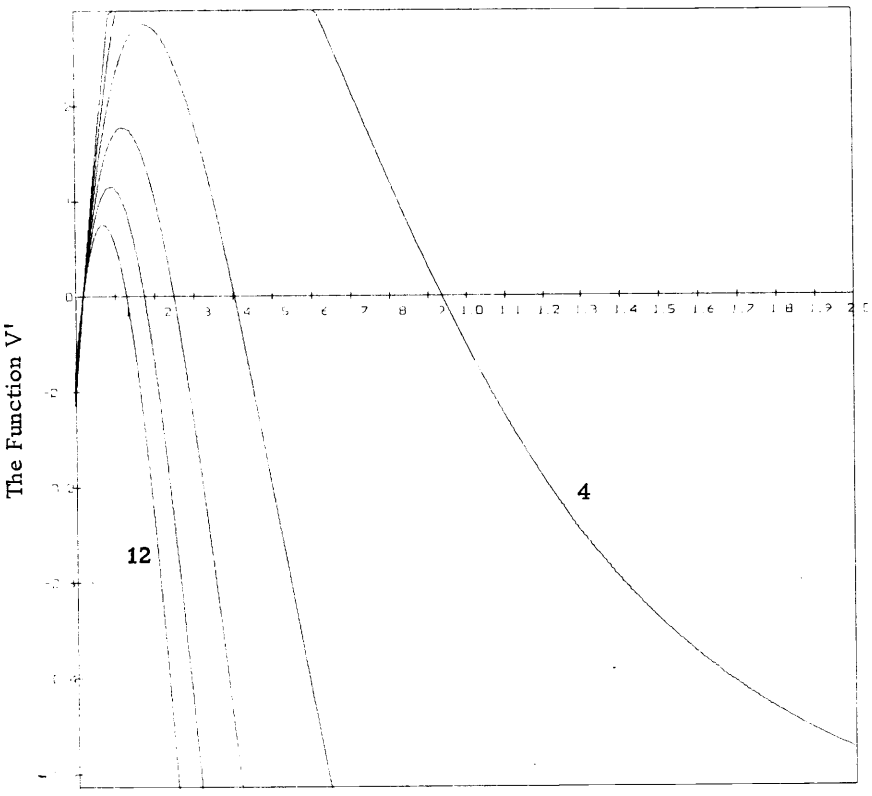

Figure A.1.k. $\lambda = .6$, $\mu = 2$, $\theta_1 = 2$, $\tau = .05$.

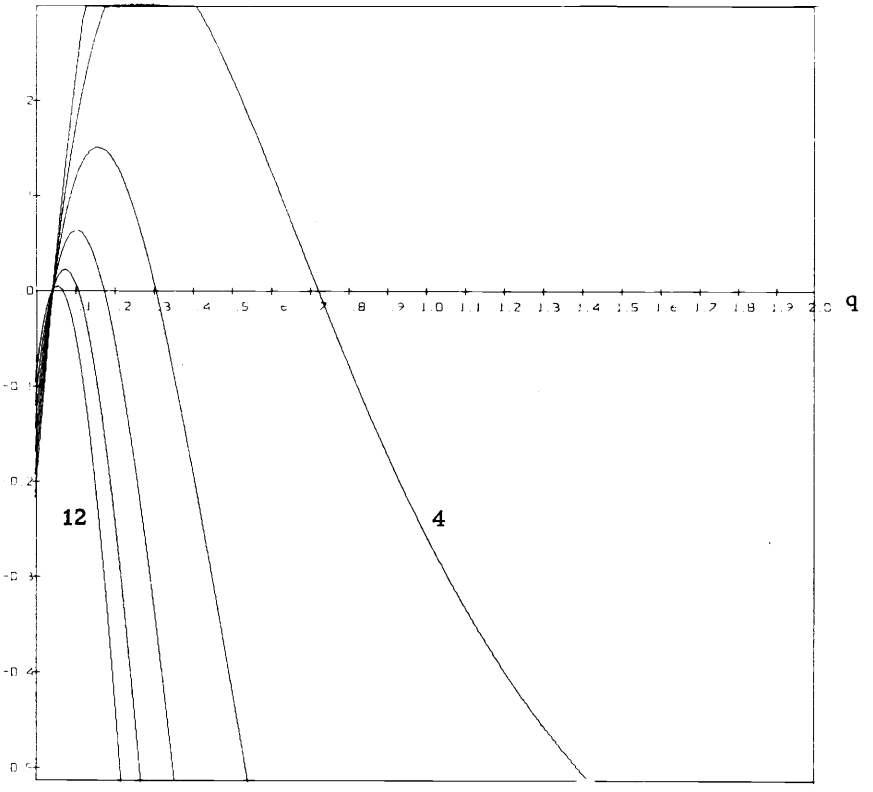

Figure A.1.l. $\lambda = .6$, $\mu = 2$, $\theta_1 = 2$, $\tau = .1$.

Figure A.1. (continued)

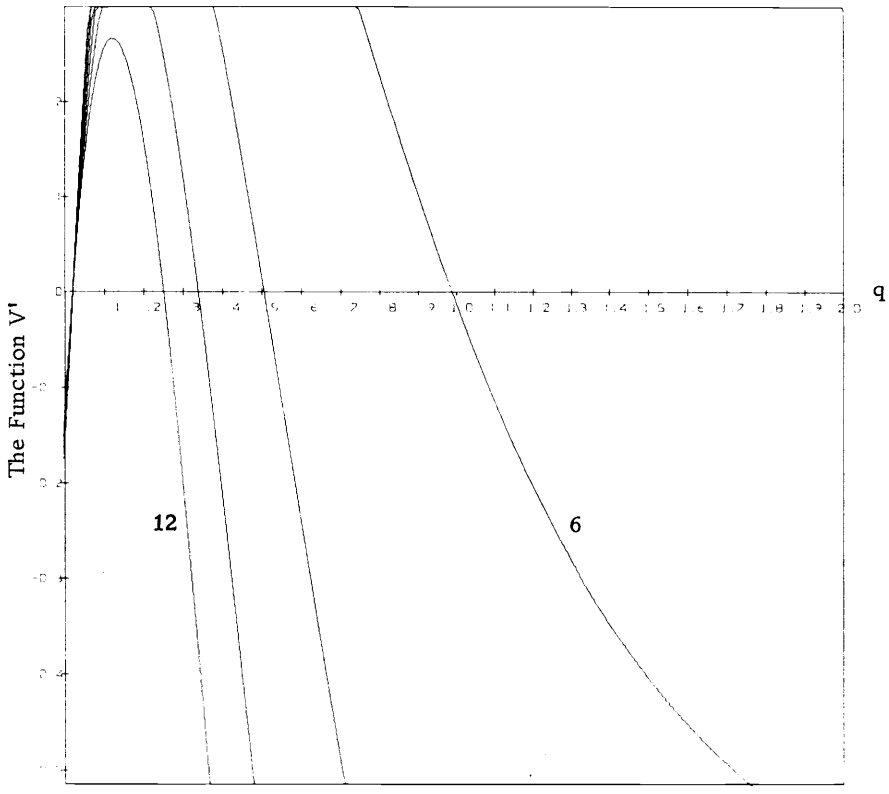

Figure A.1.m. $\lambda = .6$, $\mu = 2$, $\theta_1 = 3$, $\tau = .05$.

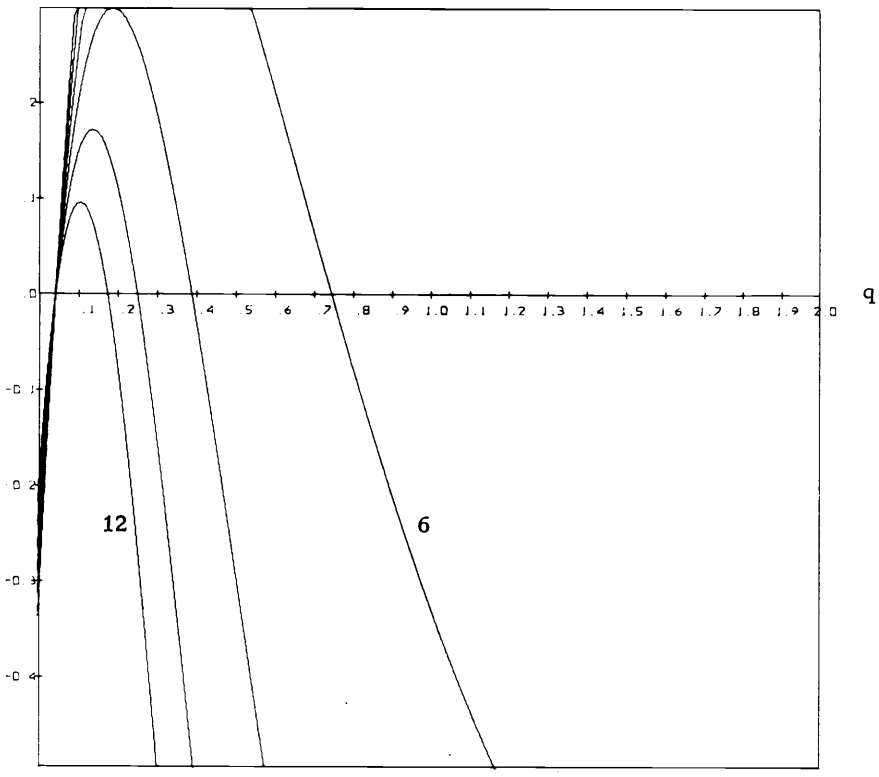

Figure A.1.n. $\lambda = .6$, $\mu = 2$, $\theta_1 = 3$, $\tau = .1$.

Figure A.1. (continued)