REPORT T-66 SEPTEMBER, 1978

COORDINATED SCIENCE LABORATORY

ADAPTIVE ALLOCATION OF DECISION MAKING RESPONSIBILITY BETWEEN HUMAN AND COMPUTER IN MULTI-TASK SITUATIONS

YEE-YEEN CHU

(NASA-CR-157937) ADAPTIVE ALLOCATION OF DECISION MAKING RESPONSIBILITY BETWEEN HUMAN AND COMPUTER IN MULTI-TASK SITUATIONS Ph.D. Thesis (Illinois Univ. at Urbana-Champaign) 150 p HC A07/MF A01 CSCL 09B G3/61 N79-12772 Unclas 38212

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

ADAPTIVE ALLOCATION OF DECISION MAKING RESPONSIBILITY BETWEEN HUMAN AND COMPUTER IN MULTI-TASK SITUATIONS

by

Yee-yeen Chu

This work was supported by the National Aeronautics and Space Administration under Grant NASA NSG-2219.

ADAPTIVE ALLOCATION OF DECISION MAKING RESPONSIBILITY BETWEEN HUMAN AND COMPUTER IN MULTI-TASK SITUATIONS

Yee-yeen Chu, Ph.D.
Coordinated Science Laboratory and
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign, 1978

Computers are increasingly being used in a number of decision making situations. As it seems reasonable to expect human and computer to eventually have overlapping abilities, adaptive allocation of responsibility may be the best mode of human-computer interaction. To give the human a coherent role in the system, it is suggested that the computer serve as a backup decision maker, accepting responsibility when human workload becomes excessive and relinquishing responsibility when workload becomes acceptable.

A queueing theory formulation of multi-task decision making is used to develop a procedure for determining when the computer should be assigned decision making responsibility. A threshold policy for turning computer on/off according to the weighted number of events present in the system is proposed. This policy minimizes event waiting cost subject to human workload constraints.

An experimental representation of a computer aided multi-task flight management situation had been developed. A computer aiding program was implemented. An experiment was conducted with a balanced design of several subject runs for different task demand levels. This was achieved using three levels of subsystem event arrival rates, three levels of control involvement, and three levels of availability of computer aiding. All experimental variables were shown to be significant in affecting most performance measures. It was seen that the computer aiding had enhanced subsystem performance as well as subjective ratings, and that the adaptive aiding policy further reduced subsystem delay.

Experimental results compared quite favorably with those from a computer simulation which employed an $(M/E_k/2):(PRP/K/K)$ queueing model. The queueing model appears to be adequate to represent the multi-task decision making situation, and to be capable of predicting the system performance such as delay time and server occupancy. This simple measure of server occupancy was found to highly correlate with the subjective effort ratings. Thus, the model has the potential for predicting human workload in multi-task situations.

ADAPTIVE ALLOCATION OF DECISION MAKING RESPONSIBILITY BETWEEN HUMAN AND COMPUTER IN MULTI-TASK SITUATIONS

BY

YEE-YEEN CHU

B.S., National Taiwan University, 1971
M.S., Clemson University, 1974

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1978

Urbana, Illinois

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude with respect and admiration to Professor Willian B. Rouse, my thesis advisor, for his guidance, encouragement and support, without which this thesis would have not been possible. I would also like to thank Professors Judith S. Liebman, William R. Perkins, Richard E. DeVor, and Rudolf G. Mortimer for their helpful comments and suggestions as committee members.

Thanks also due to my friends and fellow students, especially to Thiruvenkatasamy Govindaraj, Joel Greenstein, Heidi Neubauer, Jim Smith, and Rex Walden for their practical advice and discussions.

This research was supported by the National Aeronautics and Space Administration Grant NSG-2119.

PRECEDING PAGE BLANK NOT [illegible]

(II)

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Computers are increasingly being used in a number of decision-making situations, especially when several tasks have to be performed at the same time by a human decision maker (DM). Commercial aircraft can now, in many situations, be flown solely using computer as pilot. Industrial process can be monitored and controlled by computers. Such fast and intelligent computers can provide sound, well-evaluated decisions which may reduce system risk, human workload and errors.

On this frontier, the human has to interact with computers which are capable of processing and routing information, exerting control actions, and making choices in view of priority conflicts. The important issue arises of exactly what roles the human and computer decision makers should play as systems become increasingly automated.

The potentials for active, flexible interaction between human and computer have only recently been addressed [Rouse, 1977], [Steeb, et al., 1975]. A central issue concerns allocation of decision making responsibility between human and computer. As it seems reasonable to expect human and computer to eventually have overlapping abilities, adaptive allocation of responsibility may be the best mode of human-computer interaction. With adaptive allocation, responsibility at any particular instant will go to the decision maker most able at that moment to perform the task. Such a procedure for allocation would improve the utilization of system resources and thereby improve system performance. The emphasis of this thesis is the development of a method of adaptively allocating decision making responsibility and also, the modeling of human decision making in computer-aided multi-task situations.

1.1 Multi-Task Situations

Technology has produced a variety of machines and tools for the human to use. Higher levels of automation have continuously been introduced to further reduce the human's involvement in complex systems such as industrial process control, high performance aircraft, etc. During normal operation of these systems, monitoring the automated processes and, perhaps occasional adjustment or corrective actions, comprise a major part of the task requirements. Less frequent and more involved are situations when malfunctions arise and either backup/restart procedures or diagnosis/problem-solving processes have to be performed. Before total automation becomes socially, economically and technically feasible, these tasks are likely to be left to the human and to become a major part of his overall task. His role is therefore becoming more of a supervisor rather than of a direct controller.

Because the performance and functional demands on the system are so great, it appears that the need for human in the system, supervising and managing the operation, has not diminished. Further, his task is likely to become progressively more demanding, due to increased complexity, increased risk, and the need for more accuracy and faster response. A flight management situation to be discussed in the latter part of this thesis is an obvious example, based on projected levels of aircraft density and all-weather flight requirements [Wempe, 1974].

As another example, industrial process control traditionally allows slower response time and lower operating skills compared to flight management. However, it is getting more complicated. For economic reasons, the operator in an industrial plant is usually given the supervison of a large section of the total plant or of numerous processes. Consequently, the operator "performs a multitude of tasks in a time-varying pattern, with periods of relatively calm and other ones with frenzied activities" [Rijnsdorp and Rouse, 1977].

More specifically, a modern central control room may be equipped with dozens or hundreds of computerized CRT-displays serving as a main interface between the processes and human operator. Information such as current values or historical process data as well as alerting or emergency signals which are triggered when certain variables approach critical boundaries are presented to the human operator.

1.2 Human Decision Making

Within multi-task situations in general, it appears that an appropriate assumption is that the human has a rather narrow-band information channel capacity and if several tasks each require a substantial amount of processing capacity then the human must handle them sequentially. Furthermore, the human must devote some fraction of his total capacity to keeping track of the total parallel operation. This characterization of the human suggests an analogy between the human and a general-purpose time sharing processor.

Thus, the operator in the monitoring task can be described as in a situation where he observe one indicator at a time and progressively attends to the various indicators. A psychological stimulus-response formulation is inadequate to account for human decision making in this multi-task situation. Instead, it can be viewed as a combination of active rational information selection, manipulation, and evaluation of outputs. While the pure scanning of displays is a more or less constant fraction (approximately 0.2 seconds) of total worktime [Rijnsdorp and

Rouse, 1977], the operator may have consciously and unconsciously performed functions that include activities such as pattern recognition, prediction, induction and deduction, etc. Those skills are by nature more difficult to perform and also, for our purpose, to model. However, a well-organized task scenario as well as a structured information presentation seems to reduce this difficulty.

Senders [1964] and Smallwood [1967] have modeled human decision making in multiple process monitoring. Senders employed an information theory approach to determine how often and how long the human should sample. Smallwood proposed that the operator forms an internal model of the processes, and based on this model he directs his attention to the process with highest probability of exceeding threshold. Carbonell [1966] and Senders and Posner [1976] have proposed queueing theory approaches which relate to the multi-task formulation espoused in this thesis. These authors emphasize the monitoring of the displays, rather than perception of the displayed values and the subsequent actions of the operator.

Greenstein and Rouse [1978] propose that human decision making be modeled in terms of event detection and attention allocation. Discriminant analysis is employed to model human event detection by generating the probabilities of event occurrence as functions of features of the displayed signals. Sheridan and Tulga [1978] have modeled human attention allocation using a dynamic programming approach to determine the action sequence which maximizes the operator's earnings. These models, which emphasize the operator's task performance, may find usage in modeling coordinated human-computer decision making systems.

Rouse [1977] has addressed the issue of human-computer interaction in a multi-task situation closely related to this thesis. Queueing theory is suggested as an approach to the allocation of decision making responsibility between human and computer. The decision makers are assumed to generate, based on the displayed information, the probability that events have occurred in the tasks and the probability estimates of event arrival and action times. They then choose their actions so as to minimize an appropriate cost criterion. The simplicity in the structure of this model lends itself to flexible implementation within a variety of multi-task situations.

1.3 Workload Considerations

An important research and design issue is the correlation between human performance and workload perceived in terms of the effort expended. The workload idea employed here is derived from the concept of fraction of capacity and time-allocation. Methods that are generally used to measure this workload include: 1) secondary task performance, 2) physiological signals, and 3) subjective effort ratings.

For our purpose, it is reasonable to assume that, between boring and fatiguing, there is an acceptable workload range for the human in decision making tasks. The capacity of human information processing and decision making is seen to have an upper limit. Due to the requirement that a certain level of activity of the human decision maker should be maintained to avoid vigilance problems and loss of concentration during task execution, this capacity is also limited on the lower side as well [Pasmooij, et al., 1976]. While unexpcted task demands occasionally push the human toward the upper limit (i.e., high workload), the general trend of lowering task requirements to the mere monitoring state has furthered the possibility that the lower limit be passed. Further, wide variations of workload within the duration of a task has become one undesirable side effect of automation.

This raises the question of how to allocate decision making responsibility so as to maintain human workload within an acceptable range during task execution. For the purpose of flexible allocation with respect to this criterion, it is essential to have a workload measure that can both predict human performance and respond to changes in task demand instantaneously. A general formulation of this problem will later be discussed.

1.4 Man-Computer Interaction

The human decision maker can be described as having limited mental capabilities constrained by limited memory, inconsistent performance, and motivational factors. He, however, has remarkable perceptual capabilities [Gregory, 1966], and the flexibility to respond to unusual and unexpected situations. In general, the human's performance is slow, sloppy, but intelligent [James and Partridge, 1972]. He is sensitive to workload and is subject to several kinds of errors. The computer is characterized as fast, rigorous but rather stupid [James and Partridge, 1972]. It occasionally has hardware or software reliability problems and has limited intelligence. It appears that the members of the man-computer team have complementary talent.

The interaction of man and computer is without question a complex phenomenon. As a straightforward approach, one might allocate a fixed portion of the set of the tasks to the computer with the remainder of the set being allocated to the human. Heralding a man-computer symbiosis, Licklider [1960] has proposed guidelines for task allocation. In this symbiotic partnership, the human will set the goals, formulate the hypotheses, determine the criteria, and evaluate the results. The computer will do the routinizable work such as transform the data, simulate the mechanism and models, and implemented the results for the human decision maker.

The division of tasks is not as clear-cut, however, for decision making tasks that include computerized decision aiding systems. The rules of thumb suggested by Licklider are that man will handle the very-low-probability situations, and fill in the gaps in the problem solution or in the computer program; while the computer may serve as statistical inference, decision theory, or game theory machine, to perform elementary evaluation, diagnosis, and pattern recognition as a second role. This is the domain of the tasks for which human and computer decision makers have overlapping responsibility.

In fact, one human factors consideration in system design may be to first give the human a more coherent role (in terms of his workload, confidence, acceptance, etc). He may have to be allocated some functions which he performs at a level inferior to that of the computer. On the other hand, the computer's ability to perform intelligently appears to be evolving rather rapidly [Firschein, 1974] and an aiding program can be designed to both learn from the human and adapt to the human. Therefore, the domain of overlapping tasks seems to widen, and the allocation of responsibility in this domain seems to become a major issue of man-computer interaction.

An important issue related to the allocation of responsibility is man-computer communication. At the man-machine communication level, discussions of design of displays and input devices have been given by several authors and are summarized by Rouse [1975]. The problems of higher level communication between decision makers, such as letting human and computer know what each other is doing, without involving substantial extra workload, needs more research. It is comparatively easy, on one hand, for the computer to tell the human what it is doing. Some type of indicators visible to the human during normal scanning, could inform the human of the computer's actions and confidence in its performance, etc. On the other hand, real-time human-to-computer communication presents more difficulties. Approaches such as natural language processing [Martin, 1973], physiological EEG measures [Pinneo, 1975], and statistical model matching [Enstrom and Rouse, 1977], seem to extract sound features which can be used to characterize the human's decision making activities.

1.5 Adaptive Allocation

Given that the communication channels between human and computer have been established and that the status of decision makers and the system states could be estimated, it would then be possible to dynamically allocate functions. There are three main reasons for adaptive allocation:

1. Increased utilization of system resources. From queueing systems analysis, we know that a multiple server system where servers can move freely among queues results in much less customer waiting than would occur if servers were strictly assigned to particular queues. Thus if human and computer both are allocated full responsibility for the overlapping tasks, the events will be serviced more promptly.

2. More flexibility to cope with computer malfunctions. The possibility of the computer encountering either a hardware failure or an event whose decision making requirements exceed its abilities can never be overestimated. The sum of the probabilities of these low-probability local events may often be much too high to neglect. It would seem reasonable that the human should be allocated at least monitoring responsibility for all tasks. On the other hand, if tasks are strictly allocated, the human would not

know or attend to those operations under the computer's supervision until abnormal situations developed and placed still higher demands on the human to explore and control the subsystems. It is generally recognized that this is an inferior position for the human decision maker to the one where he would be if he had interacted with the subsystem activities. An adaptive policy offers the human more flexibility to cope with malfunction situations, and also gives grounds for training the learning machine-based decision program if applicable.

3. The human's role in the system can be substantially benefited. Since the human has potential responsibility on all tasks, he must have a perception for all tasks, and must retain the capability to override the computer's decisions when priority conflicts arise. Furthermore, in view of maintaining the human's workload, the idea of using the computer as back-up decision maker seems to be plausible and will be discussed in detail later. The adaptive policy could assure the human a coherent role in that the above considerations were taken into account.

While the adaptive policy is proposed to achieve the goal of allocating tasks to the decision maker most able to perform the task, the mechanism of allocation should be organized in such a way as to ensure human acceptance and minimum extra workload. Manual allocation control by the human at each decision epoch, either by physically allocating the task or by a threshold device set by himself, requires the human's continuous attention. This requires him to continously evaluate the system, the computer, and himself. Such a requirement might generate more workload than is acceptable. Therefore, if possible, the allocation decision should be automated, and delegated to a computerized coordinator. An algorithm has been developed to perform this role and will be discussed in the next chapter.

2. APPROACHES

2.1 Theoretical Background

In the last chapter, it was proposed that the allocation of decision making responsibility be adaptive to the status of decision makers and the system states. While the details of a solution to this problem will depend on the specific task scenario, this chapter considers a rather general, context-independent formulation of dynamic allocation in multi-task situations with somewhat concise system dynamics and clear decision goals. Reviewed in the next section are the stochastic decision and control approaches, which are suitable for describing fine-grained dynamic decision processes and are finding increasing application in modeling complex decision making situations.

2.1.1 Stochastic decision and control approaches

Since the uncertainties present in the system under consideration can be conveniently represented as stochastic processes and the two decision makers (i.e., human and computer) as two controllers with appropriate cost functions, it appears that a stochastic control formulation is plausible. The overall system can then be represented as a two-level hierarchical control structure composed of local decision making units (the human and the DM computer) and a supremal unit (the coordinator). A coordination theory has been developed by Mesarovic and his colleagues [1970], which employed supremal intervention in the local decision making units to achieve an agreement of system objectives. While an iterative procedure of coordination for on-line steady-state hierarchical system control has been developed by Findeisen and his colleagues [1978], the coordination scheme for dynamic systems needs further investigation.

If the hierarchical structure is disregarded and the competing process is emphasized, a game theoretic approach seems appropriate. A nonzero-sum two-person cooperative game problem arises when the human and the computer each have to decide how to allocate his attention based on a given level of on-line information exchanged. One difficulty associated with the use of differential game theory in the man-computer decision situation is that no clear-cut procedure exists for splitting an overall index of performance into distinct cost functionals for each DM.

In addition, the man and computer are, in general, in a dynamic decision team, in which the information obtained by one DM is also affected by what the other DM has done. For simplicity, we can reasonably assume that either there is perfect communication or that the system allows for one step communication delay which enables the reduction of a dynamic problem to a static one [Ho and Chu, 1971]. Under these assumptions, it appears that we may model human and computer decision making behavior separately (a decentralized structure of information and control) while seeking the same goal (centralized structural optimization).

In each of the local decision making units, the problem is to develop a policy for performing experiments (i.e., monitoring) and then allocating action resources on the basis of the outcome such that a performance index is optimal. One important issue is that optimal stochastic control requires the solution of stochastic dynamic programming equations which are infinite dimensional. An approximation solution in this case has been proposed by Bar-Shalom, Larson, and Grossberg [1974].

Another issue is that the separation principle does not hold for two controllers with different information sets [Chong and Athans, 1971]. Aoki and Toda [1975] consider a two-level hierarchical decision with decentralized information system and suggest a certainty equivalence control which gives a suboptimal decision algorithm with learning (Bayesian estimates). Both approaches require too much computation to be implemented in an on-line real-time multi-task environment with reasonable implementation cost.

A more realistic representation of the supervisor-regulator-subsystem decision unit is given by Forestier and Varaiya [1978] employing a two-layer feedback control of Markov subsystem process. At the lower layer a regulator continuously monitors the subsystem. When the state of the subsystem reaches extreme or boundary values, the supervisor at the higher layer intervenes to reset the regulator. The study showed that while the supervisor needed to intervene only at reset instants, the structure would require the supervisor to obtain complete knowledge of the lower layer transition probabilities and costs. Further study is required to determine an applicable adaptive control policy based on this structure.

Hsuan and Shaw [1976] and Sworder and Kazangey [1972] consider quadratic linear stochastic random jump process within control and dynamic repair situations. If we want to penalize deviations of task states from some desired values, then we might formulate the criterion so as to allow for the use of their results. On the other hand, if our main interest is to minimize the average delay of servicing events or to appropriately allocate workload, then a queueing theory approach seems appropriate.

2.1.2 Queueing theory approaches

When the time expenditure in the various decision tasks is of major concern, the multi-task decision making system may be considered as a queueing system with two servers (human and computer) and K classes of customers. Thus, we have simplified the problem of allocating decision making responsibility to be one of determining who serves a particular customer or, equivalently, which server the arriving customer should be directed to.

In a system similar to our multi-task situation, Carbonell [1966,1968] presents a queueing model of many-instrument visual sampling. The model is based on the concept of the different instruments competing for the attention of the human. At each sampling instant the decision as to what instrument to look at is based on the combined effect of both the probability and the cost of exceeding the threshold. This model has been validated for human visual sampling. A queueing model of the human decision maker that emphasizes the important aspects of multi-task decision making noted in earlier discussions has been developed by Walden and Rouse [1977].

Man [1973], and Hsuan and Shaw [1976] have separately presented the stochastic optimal strategies for arrival rate regulated and service rate controlled systems with time varying input traffic demand. A set of continuous time state-space differential equations is derived, the maximal principle is applied, and a two point boundary value problem is obtained. Due to the formidable complexity in computation, the author suggests that a bang-bang fixed threshold policy would be an acceptable suboptimal solution.

Using a queueing system framework, the technique of Markov decision processes has been applied by many researchers to solve the queueing control problem. We will first present results for optimal control of queues. A more thorough review of literature with emphasis on the dynamic control of queues using service variables, arrival variables, and priority disciplines is given by Chu [1976].

Heyman [1968] considers the problem of controlling a queueing system with Poisson arrivals, general service time distribution and single server (M/G/1) by turning the service mechanism on when a customer arrives or off when a customer leaves. He shows that the optimal stationary policy which minimizes linear average or discounted cost over an infinite horizon has a simple critical number characterization: (M,m). This (M,m) policy is to provide no service if the system size N (i.e., number of customers in the queue) is m or less, and to turn the server on when the size N is greater than M (referred to later as the

N-policy). The cost incurred includes waiting cost, running cost, and switching cost. This result is quite similar to those obtained from inventory control theory.

Bell [1971,1973] extends the result to an M/G/1 nonpreemptive priority queue and proves the existence of an optimal average cost policy of the (a,b,c) type for two priority classes. This optimal policy is never to turn the server off or to turn the server off only when the system is empty and to turn the server on the first time that $a\ n_1 + b\ n_2 \geq c$, where n_1 and n_2 are the number of class 1 customers and class 2 customers in the system. For the general K priority classes, the optimal control actions are simply characterized by the (K-1)-dimensional hyperplane of the form: $a_1n_1 + a_2n_2 +\ldots+ a_Kn_K = c$. This result will be utilized in the later discussion.

Balachandran [1973] has considered the same on-off policy with control measures determined by the unfinished work D in the system (hence the approach requires that the service time be known for customers in the system). This D-policy is later proven by Balachandran [1975] to be superior to the usual N-policy.

Shaw [1972,1976] presents results for optimal assignment of servers or rejection (detour) of customers on the basis of the arriving customer's waiting time W in queue. He shows that the optimal customer diversion to minimize total delay to accepted and detoured customers is of the control limit form for the M/G/1 case. While the number in the queue in an N-policy is easier to measure than waiting time, it is useful to know what performance could be achieved by this W-policy if a greater amount of information is available.

To employ the N-, D-, or W-policies, an idle server must constantly monitor the queue for an arrival. When this situation cannot occur, Heyman [1977] proposes the T-policy where the server scans the queue T time units after the end of the last busy period to determine if customers are present. He shows that under the optimal T-policy, the corresponding N-policy is optimal. He also proves that the optimal N-policy is always better than the optimal T-policy, which seems intuitively to be correct.

Concerning the value of information and preemption, one may expect to do a better job if one has better information about the likely processing time of a job or if one is allowed to preempt a job which is in process. Schrage [1975] surveys the analytical results in scheduling under uncertainty. He concludes that the optimal nonpreemptive sequencing strategy for a linear cost criterion is to employ the first come first serve (FCFS.) disciplines under no information situations and to employ the shortest processing time first (SPT) disciplines under full or partial information. In the preemptive disciplines, the shortest remaining processing time first (SRPT) disciplines are employed in substitution for the SPT for the system to be optimal.

On the priority assignment among K tasks, a useful result is given by Cox and Smith [1960] based on the service rate μ_k and the waiting cost c_k for an M/G/1 system: of all the nonpreemptive work-conserving stationary policies, the head-of-the-line discipline with priority assigned to the class k customer with higher $\mu_k c_k$ product is that which minimizes the average waiting cost. However, this simple solution does not hold for a system with a finite queue. As shown by Mova and Ponomorenko [1974], the optimal priority assignment depends on the time-varying system configuration (i.e., specific pattern of the waiting line).

For systems with known parameters, there are well-known methods for dynamic priority assignment. The standard procedure is to set up the probability equation (Markov process, stationary regime), to apply the necessary condition for the principle of optimality and then to solve either a linear programming or a dynamic programming problem [Mova and Ponomarenko, 1974, Nazarov, 1976].

However, if there are unknown parameters such as the arrival rates or service rates, the necessity arises of considering adaptive systems both neutral (i.e., anticipated future information is of no value) as well as with dual control (i.e., anticipated future information requires simultaneous solution of the control and estimation problems). Nazarov and Terpugov [1976] propose a variant of the Bayesian approach for finding an adaptive control on an unknown parameter with given prior distribution. Assuming that there exists the possibility of obtaining additional information through experimentation to supplement prior judgement, Bagchi and Cunningham [1972] show how a statistical decision theory approach may be gainfully applied to handle the uncertainty of parameters pertaining to the optimal design of queueing systems.

The above literature considered queueing systems with independent identically distributed arrivals. In most real situations, however, the inputs are correlated. Gopinath and Morrison [1977] presented the analysis of single server queues with inputs represented as a sum of moving averages. The analysis of the situations where events arrival are correlated deserves further effort.

Summarizing the literature, there is certainly an appreciation for the complexity involved in the stochastic control approach as well as the Markov decision approach for representing multi-task decision making situations. No results have been found concerning the control of queues of two servers, and of finite waiting places. In fact, the multi-task decision making situation discussed here is much more complex than any results available in the literature.

There are, however, several useful suggestions: the threshold policies probably are acceptable suboptimal solutions; the better information the better control; the optimal priority assignment of tasks could deviate from a static one such as the μc solution; and further adaption to unknown parameters can make the system less vulnerable to uncertainty.

2.2 The Proposed Approach

In view of the complex task situations under consideration, it appears both natural and appealing that the system lends itself to a two-level hierarchical structure with the top level coordination between human and computer being our main concern. Considering a task domain where the computer is employed as back-up decision maker, the problem is further simplified by assuming that the coordinator has all the information needed and that both DM have a common, centralized system goal. Then the simplified coordination problem, to be first investigated here, becomes one of self-organizing on the part of computer DM: When should the computer request and relinquish responsibility?

For the low-level decision making within a multi-task situation, the human DM is assumed to appropriately allocate his attention and effort among the tasks. He is assumed to employ a quasi-optimal decision making strategy for scanning displays and allocating attention. This is based on the assumptions that the tasks are independent and that events either unequivocally or progressively present themselves. The human DM scans the task display in order of decreasing priority at a given rate. He then performs the first task for which he perceives some action-evoking events. The computer is assumed to adopt the same strategy either by being hard-wired or learning from the human DM. More specifically, the basic description of a multi-task decision making situation is as follows:

1. The multi-task situation can be classified as K independent tasks characterized by a set of state vectors, x_k, $k=1,2, \ldots, K$.

2. The prior statistics for the observations of system state are given (for example, the joint distributions of the presence of events, interarrival times, and action times etc.).

3. The DM scans the task display in order of decreasing priority at a given rate. He then performs the first task for which he perceives some action-evoking events.

4. The performance index (cost function) is given as an average waiting cost, weighted according to the importance of each task.

5. The DM has given probabilities of making false alarms, missed events, and incorrect actions, which may depend on other situation parameters such as the overall arrival rate.

A simulation of this basic multi-task situation is discussed by Rouse [1977]. Poisson arrivals and exponential service distributions were assumed. Two results are important to note. First, the false alarms were more detrimental to performance than the missed events. Second, the average delay increased quickly as probability of conflict increased. The degree that human and computer know about each other's action was shown to be important. Using the multi-task context described, we now want to consider the question of when the computer should request and relinquish responsibility for a portion of the tasks.

2.2.1 The optimal adaptive policy

We will consider the design of an adaptive computer decision making system for the multi-task situation discussed earlier. Assuming there are K processes, at epoch i process k can be characterized by a state vector $x_k(i)$, $k=1,2, \ldots,K$, while the decision maker j is characterized by his observation of the state,

$$z^j = (z^j_1, z^j_2, \ldots, z^j_K),$$

his perception of event occurrence,

$$p^j(.|z^j) = p^j(e_1, e_2, \ldots, e_K|z^j),$$

his perception of event interarrival time,

$$f^j(.|z^j) = f^j(t_{e_1}, t_{e_2}, \ldots, t_{e_K}|z^j),$$

and his perception of event service time,

$$g^j(.|z^j) = g^j(t_{s_1}, t_{s_2}, \ldots, t_{s_K}|z^j).$$

Combining the above information and the system performance criterion allows the decision makers to determine their strategies [Rouse, 1977].

It is impossible to give a universal performance criterion. In terms of a stationary expected cost structure, a convenient measure seems to include waiting cost, service cost, and switching cost. By assigning relative waiting cost rates and minimizing the average waiting cost, one can take into account delay as well as relative risks.

While one might regard the cost of temporarily diminished capacity of the CPU as the cost for switching on the computer, we will assume it to be negligible. Thus, the optimal policy seems to be to have the computer on all the time. However, even without explicit costs, the possibility of server errors such as false alarms and incorrect actions as well as degradation of service rate will yield effective costs and thereby may lead to non-trivial solutions. This allows the use of a modest and analytically appealing formulation for expected cost such as

$$E[C] = E[\ c_1 w_1 + c_2 w_2 + \ldots + c_K w_K\ ],$$

where $E[\cdot]$ denotes the expected value, w_k and c_k are the delay and the cost per unit delay time of service to process k.

Human workload as it affects performance degradation (e.g. decrease in service rate or increase in service error) is an important issue. Considerable human factors engineering has been aimed at reducing workload to avoid an overloaded condition. But there is evidence of vigilance and warm-up decrement on sustained manual operation, and issues of new workload measurement methods are under investigation [Verplank, 1977; Moray, 1978]. If there is an optimal workload that sustains performance on long tasks, we may want to seek a policy for computer aiding such that an optimal workload is achieved.

Thus, we have two performance criteria and, it is quite likely that the optimal workload solution does not coincide with the minimum waiting time solution and one must trade between these two criteria. One way to avoid this difficulty is to assume that human performance degradation can be represented by the increase in his chance of making errors or perhaps a decrease in his service rate. Thus, we assume a functional relationship between error probabilities and/or service rate and workload. By appending this functional constraint to the minimum waiting time formulation in a way similar to the method of Lagrange multiplier, the policy is then forced to take the human workload into consideration [Chu, 1977]. Another alternative is simply to assume a workload interval which is acceptable for a specific task, and to minimize the average waiting time subject to this workload contraint.

Also, it may be found that the relationship between false alarms and human workload is not strong enough to dictate giving a higher workload to the human purely on the basis of waiting time. In this case, if workload is the primary consideration, then the computer's threshold should adapt to the subsystem arrival rate since it is the primary cause of changes in workload.

In view of the above theoretical background and the complex task situations, we will advocate the use of the stationary expected cost optimal policy for computer on-off of the following form: turn the computer on at arrival epoch when $N = c_1n_1 + c_2n_2 + \ldots + c_Kn_K \geq M$ and turn it off when $N \leq m$, where M, m, c_1, c_2, $\ldots$, c_K are non-negative constants and $n_k=\emptyset$ indicates that there is no event in process k, while $n_k=1$ indicates that there is an event. The c_k are chosen according to the relative priorities of events. Bell's results [1973] imply that the c_k here happpen to be the same as the assigned constant cost rates c_k for single server, two priority process situation. This choice of N-policy (which depends on the number of customers present) is based on the ease with which it can be measured, its responsiveness to the variation of arrival rate and service rate and the fact that considerable literature suggests this measure.

The optimal threshold policy (i.e., M and m) thus obtained should vary as the system variables vary. The sources of variation include: 1) traffic demand (arrival rates), 2) server performance and task complexity (task involvement, service rates and probabilities of error), 3) system and performance uncertainties (unidentified parameters). An approach to implementing the adaptive optimal policy is to set up a table of stationary control policies off-line and to implement a table look-up along with on-line identification and estimation of system variables. In the next two sections, we will discuss two approaches for obtaining optimal stationary policies.

2.2.2 Determination of the optimal thresholds

The control policy to be discussed is to turn the computer on at arrival epochs, if the total number N of events in the system is greater than or equal to M and to turn the computer off, at completion epochs of computer service, when N is smaller than or equal to m. Assume an (M/M/2) queue with general priority discipline, finite K waiting places, and finite population K: (GD/K/K). In such a case, the k-process cannot go 'down' more than once at the same time and thus, the total arrival rate is $\lambda = \sum_{\kappa=1}^{K} \lambda_{\kappa}$, when $n_k=0$ for all k. An analytical approach is to write the steady-state balance equations [White, Schmidt, and Bennett, 1975] for given M and m. The steady state probability vector which characterizes system state is given by

$$\underline{p} = [p_{n_1 n_2 \ldots n_K ij}],$$

where $n_k=1$ indicates a k-event in the system; $i=\emptyset$, or $j=\emptyset$ indicates idleness, and $i=1$ or $j=1$ indicates business for the appropriate server. The sequence $n_1 \ldots n_K$ can take on 2^K unique patterns of 1s and Øs. The state equations can be written, using the rate in equals rate out approach, in the following form (details are shown in Appendix I)

$$\underline{A}\ \underline{p} = \underline{b},$$

where $\underline{A}$ is a $2^{K+2} x 2^{K+2}$ matrix and $\underline{b}$ is a 2^{K+2} vector which is determined with the state-dependent arrival rates and service rates given. For a simple six process, no server error problem, the solution of state probabilities requires an inversion of a 256x256 matrix.

With the state probabilities defined, we can calculate operating characteristics such as the average waiting time W and server utilization ρ_1, ρ_2. After deleting all unreachable states for an (M,m) policy, the matrix dimension can be reduced to an order of 2^N+2^M-1. However, if we are to further allow for two types of server errors (namely, the false alarm for i,j=2, and the incorrect action for i,j=3), we will find that the matrix to be inverted is of order

$9(2^N-1)+3(2^M)-1$. Then a modest six process, M=6 problem requires inversion of a 758x758 matrix.

The difficulty of the matrix inversion could probably be eliminated by taking advantage of the matrix being fairly sparse [Duff, 1977]. However, a more important difficulty arises when unequal costs of delay (among processes) is considered. This requires that we determine the particular patterns of $n_1 n_2 \ldots n_K$ that will exceed that threshold. One then must rewrite the balance equations for each set of patterns.

Thus, while an analytical solution to optimal control of the (M/M/2):(GD/K/K) queue is possible, it is unreasonably cumbersome for the situations which we wish to consider.

2.2.3 Simulation approaches

Because of the complexity of the analytical solution, a simulation approach may be adopted to determine the optimal stationary policy. A FORTRAN simulation program based on one discussed by Rouse [1977] was developed for the computer-aided situations. Using an activity scanning approach to simulate an (M/G/2):(GD/K/K) queue, the program maintained separate process mechanisms for each individual task, including false alarms for each decision maker. Among the assumptions of the program were fixed priorities among the tasks and constant probabilities for incorrect actions and missed events for each decision maker, although further generalizations are not too difficult. Preemption of the high priority tasks over low priority ones and override of the human over computer were also possible.

There were three classes of input variables in the simulation. The first class included process arrival rates, service rates, and waiting cost rates for subsystem processes. The second class of variables were those specific to the decision makers: the probabilities of incorrect actions and missed events, the false alarm arrival rates and service rates, scan times, task switching times, and computer on-off switching times, etc. The third class of variables included the control limits, M and m. The simulation output supplied statistics for the operational characteristics of interest such as delay time, server occupancy etc. (Program structure is shown in Appendix II.)

Program validity was tested by comparing the resulting average waiting time (for the cases of equal costs of delay, single and double server without error) with that obtained from an analytical solution for an (M/M/c):(GD/K/K) queue

[White, Schmidt, and Bennett, 1975]. For all the cases tested, the hypothesis that the two sets of solutions have the same mean waiting time was not rejected at the 5% significance level.

3. FLIGHT MANAGEMENT AS AN EXAMPLE

3.1 Flight Management

As aircraft become more complicated and greater demands and better performance are being required of the pilot, the development of automated airborne systems to share the tasks of piloting an airplane becomes increasing attractive. Advances in electronics and computer technology have made this approach both feasible and promising. Progress in sophisticated cockpit design and growth in avionic computer systems reflect the trend.

As an example, McDonnell Douglas has introduced a digital flight guidance system and category 3A autolanding system with 50 foot decision height in the DC-9 Super 80 to reduce pilot workload [Smith, 1978]. Included in the system are Sperry dual digital computers to control autopilot, flight directors, speed control, and autothrottles. The French Air's A-300 all weather autolanding system is another example which is capable of performing category 3B takeoff and landings on a daily basis [Ropelewski, 1978]. The system allows takeoff with runway visibility as low as 330 feet and a 25 foot decision height landing with 400 foot RVR.

The automated navigation system of British Airways employs a Control and Display Unit (CDU) supplemented with an Electronic Automatic Chart System (EACS) to interface the pilot and the navigation computer [ARMA, 1977]. The CDU is used to insert navigation information from the pilot into the computer store, and to annunciate system status and malfunctions, whereas the EACS generates a cockpit map display which provides a presentation of aircraft position and heading moving against a background map showing appropriate navigation data. Then the pilot is allowed to plan flight paths by inserting waypoints, editing a route, or changing marker points.

The airborne traffic situation display system developed by Connelly [1977] presents an integrated traffic, map and weather information to allow the pilot a greater degree of participation in the air traffic control process. Connelly also predicts that the key element in post-1985 period is the development of a modularly expandable avionic device that can provide navigation, collision avoidance and communication functions.

Equipped with autopilot and subsystem computers performing automatic navigation, guidance, energy calculations, flight planning, information display, etc., the next-generation of aircraft are quite likely to be capable of carrying out all phases of flight automatically. However, the human pilot is likely to remain a part of the system to cope with unpredicted or failure situations for which automation may be economically or politically infeasible. The pilot's roll then is changing from one of controller to one of supervisor and manager, responsible for monitoring, planning and decision making.

The pilot as the airborne system manager has responsibility to monitor the aircraft subsystems such as navigation, guidance, etc. as well as the autopilot and to detect possible hardware failures and potential hazards. He must constantly respond to action-evoking events such as: to communicate information, to change aircraft configuration and to reduce 4-D accuracy errors. He is also required to respond to unexpected events such as a change in flight plan, to establish the backup mode, and to declare emergencies, etc. [Wempe, 1974]. The pilot is in a multi-task situation.

If the pilot perceives an irregularity in one of the subsystems, he may seek more detailed information through either the on-board information system or actual sensor readings. Or, if he considers the irregularity to be minor, he may decide to continue his monitoring for higher priority events. There may also be autopilot malfunctions or sudden changes requiring the pilot to take charge of flight control. A proper representation of information through a flight map display indicating the continuous functioning of automatic control may help to ensure his remaining alert and responding quickly.

As described above, the automated system can normally take charge of the whole system except during critical situations such as when the system is suffering from a malfunction. Or a high-workload situation may develop when the aircraft is close to the ground and a high level of pilot activity is required. In all of these situations, the pilot is more than usually busy and further assistance of a computer would be most useful.

The recent development of fast and intelligent computer systems presents the potential for providing sound, well-evaluated airborne decisions which could reduce system risk, pilot workload and errors. While the computer as a decision maker is basically an implemented set of algorithms, adaptation and learning is possible. It is reasonable to expect that this evolving "intelligent" computer may be employed as the supervisor to the subsystem computers, taking charge of the tasks within its decision capability. The pilot and the computer thus have comparable abilities and overlapping responsibilities in performing these tasks.

3.2 An Experimental Situation

Two experiments are to be discussed here. A brief review is given of an experiment previously reported by Walden and Rouse [1977] investigating pilot decision making in an unaided situation. The second experiment, considering the computer aiding and autopilot malfunction situations, employs basically an outgrowth of the experimental representation used in the previous experiment.

The experimental situation developed earlier [Rouse, Chu, and Walden, 1976] used a PDP-11 driven CRT graphic system to represent a cockpit-like display to an experimental subject. (The experimental apparatus and simulation software used are described in Appendix III.) The display shown in Figure 3-1 included standard aircraft instruments such as artificial horizon, altimeter, heading and airspeed indicators. Also displayed was a flight map which indicated the airplane's position relative to the course to be followed. A small circle moved along the mapped course indicating the position the aircraft should have for it to be on schedule.

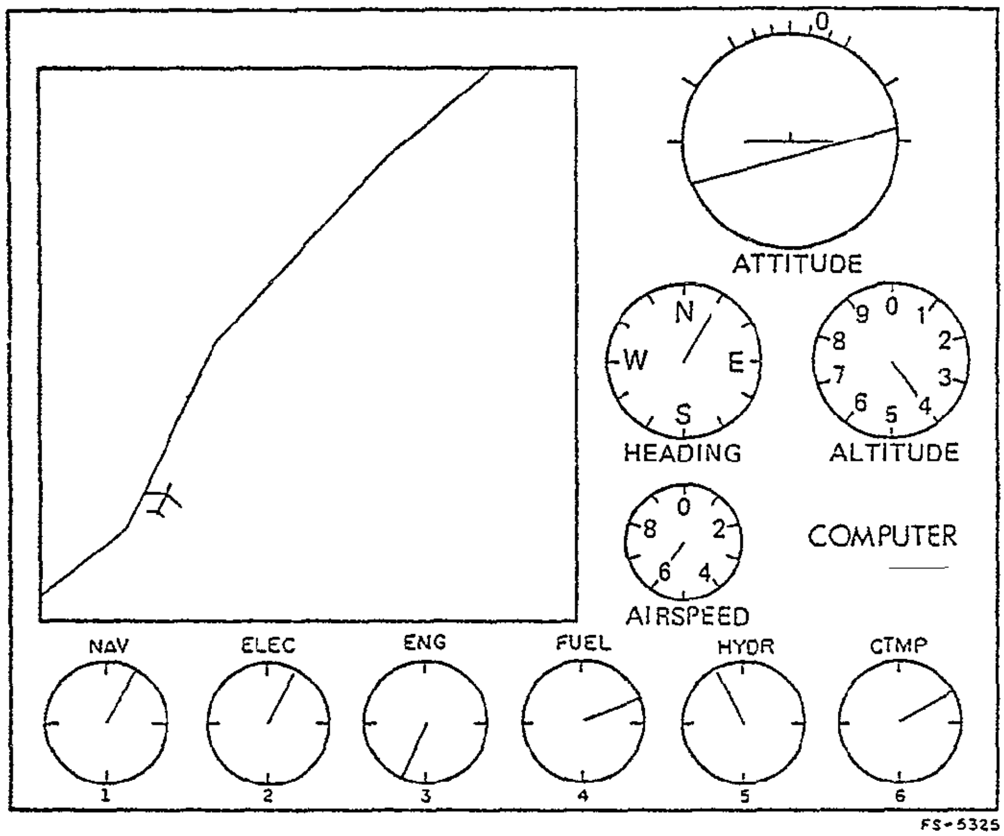

Figure 3-1. The flight management situation.

In the manual control mode, the pilot controlled the pitch and roll of Boeing 707 aircraft dynamics with a joystick. (The aircraft dynamics was taken from Blakelock [1965] and is described in Appendix IV.) Another control stick regulated the airspeed. The pilot's control task was to fly the airplane along the mapped route while maintaining a fixed altitude and stable pitch and roll attitude.

Below the map were the subsystem dials that represented the numerous aircraft subsystems which the pilot monitored for possible action-evoking events. Upon detecting an event (represented by the pointer pointing downward as shown for the engine subsystem in Figure 3-1) to which he wished to respond, the subject selected that subsystem via a 4x3 keyboard. The display shown in Figure 3-2 then appeared. This represented the first level of a check list-like tree associated with the subsystem of interest. He then searched for a branch labeled with a zero and selected the branch with his keyboard. After completing the last level of the tree, the action was completed and the display shown in Figure 3-1 returned, with the subsystem information or diagnostic check complete.

The subsystem events were scheduled to arrive according to a prescribed Poisson distribution. Events of different subsystems arrived independently with fixed priority. The

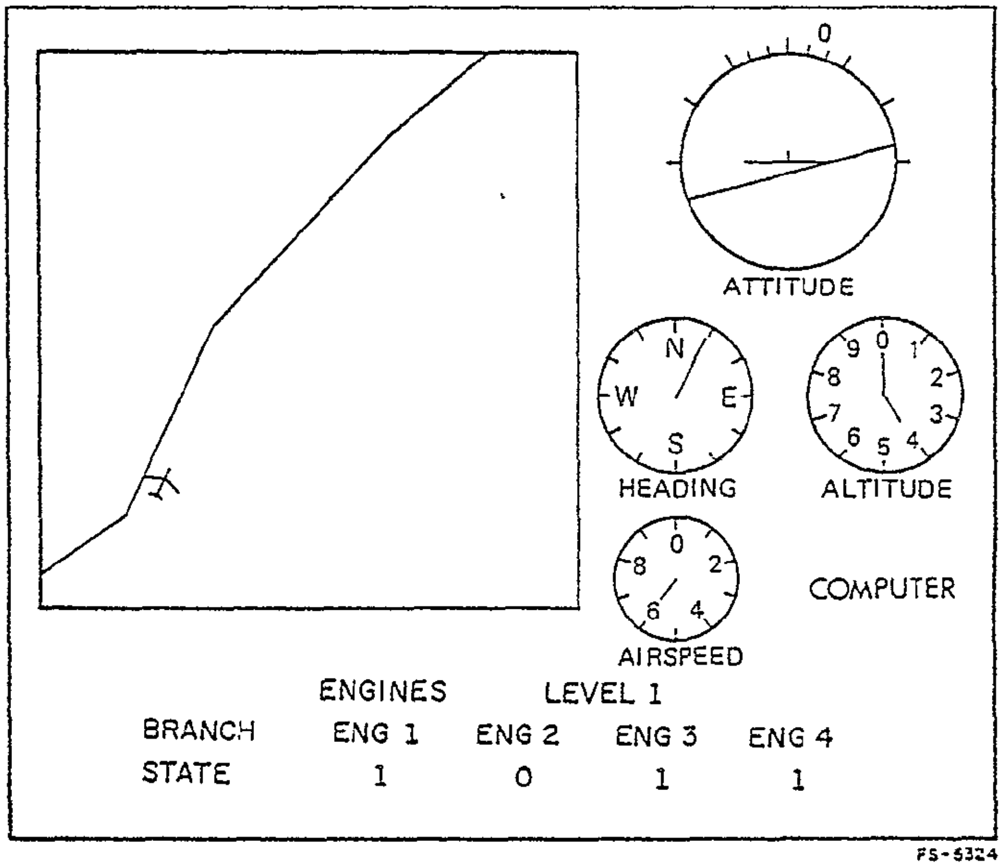

Figure 3-2. Display when pilot had reacted to an event in engine subsystem.

subjects were advised to place a high priority on the control tasks than on subsystem tasks; and within subsystem tasks, priority decreased from left to right. For example, the navigation subsystem was the most important while the cabin temperature subsystem was the least.

Using the experimental situation, an experiment was performed by Walden to study unaided pilot decision making strategies and the resulting performance. The two independent variables in the experiment were the inter-arrival time of subsystem events and the difficulty of the flight path. The results showed that, while average waiting time increased with subsystem event arrival rate, the average service time appeared to be independent of subsystem arrival rate. The waiting time was also shown to increase as the control task was added. This effect was only a function of the mere presence of the control task, rather than the control task difficulty. Incorrect actions in servicing subsystems tended to increase with subsystem arrival rate, but showed no consistent variation with control task difficulty. False alarms, however, tended to occur more frequently with the easier control task and lower subsystem arrival rate. This presented evidence of performance degradation under low workload situations.

The data collected was used in the queueing model of pilot decision making in an unaided monitoring and control situation. The model gave a reasonable prediction of pilot performance in performing subsystem tasks, suggesting that it was an adequate description of pilot decision making in the given situation and that a similar model would be useful in the adaptive aiding system.

Based on the experimental representation discussed above, a new experimental situation for adaptive aiding was developed with the aiding program (i.e., the computer decision maker) and the coordinator program (i.e., the on-off algorithm) added to the original system. Issues concerning the capability of the computer to perform the subsystem tasks, the communication linkage between the pilot and the computer, and the activities of the coordinator deserve further discussion.

The computer is assumed to be able to perform monitoring and diagnostic check procedures using information from channels linked with subsystem computers and from the data links. It makes no errors such as false alarms, missed events, or incorrect actions after it gains confidence in performing the task. The detection and service times are assumed constant. As for the service discipline among the subsystems, the computer employs the same priority rule as that used by the pilot. To be consistent in its back-up role, the computer adapts itself to the pilot and avoids interference with him. To this end, the pilot is allowed to override any decision the computer has made.

Without knowing what each other is doing, the pilot and the computer may compete for the same task or resource. The prospect of conflict between the two is highly undesirable, since, it simply causes confusion and also results in higher workload and degraded performance. The question as to how to design effective communication links without increasing the pilot's workload becomes important.

To inform the pilot of the computer's action, a succinctly displayed computer status indicator on or near the subsystem displays would seem to be satisfactory. Relevant information, if needed by the pilot for further details, may be structured into the hierarchical check-list procedure. In the experimental situation shown in Figure 3-3, The 'NAV' symbol over the navigation dial flashed if the computer decided (when the threshold was exceeded) that an event had occurred and was waiting to be serviced in the navigation system. The purpose of this indicator was to inform the pilot that he could take charge of the navigation system and the computer would take some other responsibility to avoid interference; otherwise, the symbol would continue

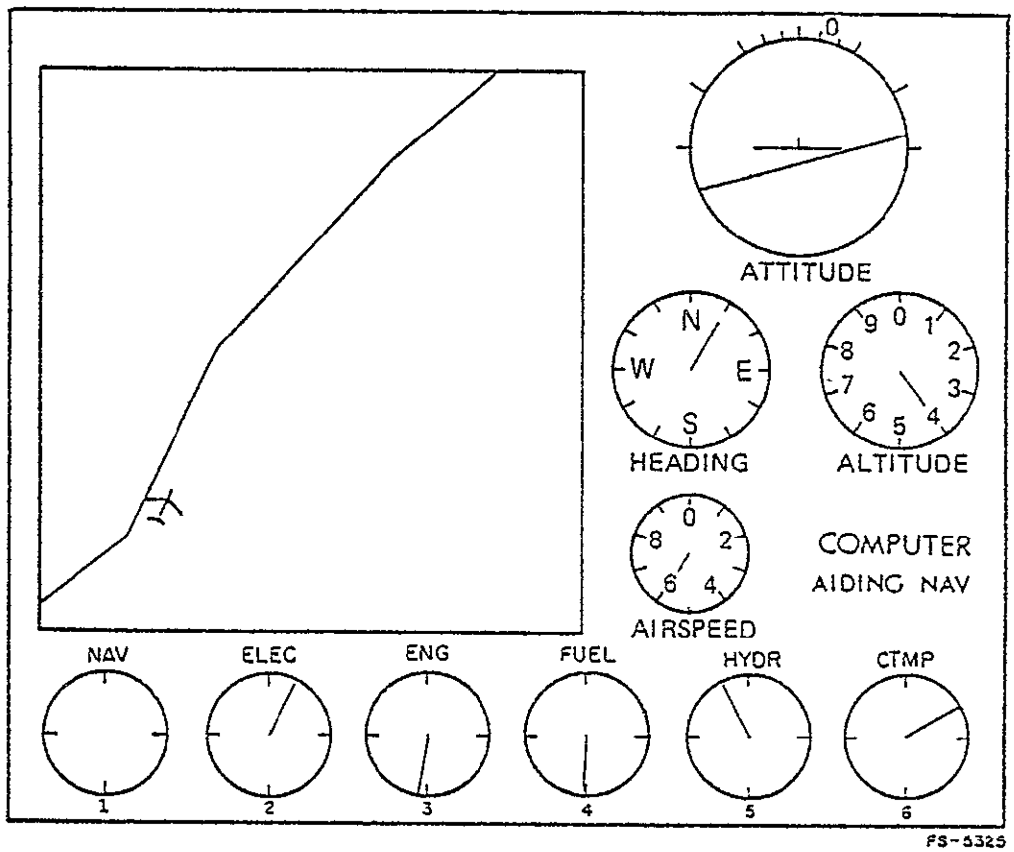

Figure 3-3. Display when the computer is servicing navigation system.

to flash for a total period of four seconds until the computer started interacting with the navigation system, resulting in a dim indicator showing in the navigation dial. If the pilot was in the middle of performing some other subsystem check procedure, say within the engine system, he would not see the flashing 'NAV' symbol over the navigation dial. The status of the computer was then shown on the lower right hand corner of the CRT by an 'AIDING NAV' symbol (flashing during the interval of possible pilot preemption), if the computer was awaiting preemption or interacting with the navigation subsystem. This computer status area was blank if the computer was not actively involved in the subsystems.

Airborne pilot-to-computer communication is, in general, more complicated. Problems involved include estimating and processing signals as well as matching or recognizing system states. For the purpose of the experiment reported here, however, the communication channel from the pilot to subsystems was narrowly defined. For our experimental situation, these included the keyboard input and stick response sampling (through an A/D converter). These channels provided the monitoring computer a way of determining if the pilot was interacting with any portion of the system. If a number had been received through the keyboard, and the checklist was being processed then the pilot had to be performing a subsystem task. The deviation of stick from normal position revealed that the pilot was performing the control task.

While the computer had to constantly check the pilot's actions to avoid conflicts, the coordinator had to synchronously check the subsystem states to determine if there was any system change. The decision epoch was when an event arrival or departure occurred. Then the coordinator calculated both the weighted sum of events and the threshold. The criterion discussed earlier was used to determine if the computer was to be turned on at the arrival epoch or to be turned off at completion epoch.

Data, sampled synchronously (twice per second), included subsystem status and states, autopilot status, aircraft dynamic variables, stick and keyboard responses, computer status and the threshold values.

3.3 Formulation of the Flight Management Situation

We have proposed that responsibilities not be strictly assigned to each decision maker. Instead, allocation should adapt to the state of the aircraft and the state of the pilot [Chu and Rouse, 1977]. Further, to retain a coherent role, the pilot should be given overall responsibility for the whole aircraft while the computer would enable the pilot to avoid having to continually exercise all of these responsibilities. On one hand, it may not be appropriate for the computer to make the vital, final judgement where losses may extend beyond the point of recovery. On the other hand, there may be vigilance problems and the pilot's performance may degrade. This leads to the idea of utilizing the computer as a backup for the pilot. The allocation problem becomes one of deciding when the computer should request and relinquish responsibility.

Given these descriptions, we will explore several issues concerned with pilot decision making in computer-aided flight management situations. Is system performance enhanced by computer aiding? How effective are different aiding policies? How does the pilot feel about aiding? Is his role or performance affected? To investigate the feasibility of the approach, and to predict the effects of numerous system variables and aiding policies, a queueing formulation of multi-task decision making with two servers (the pilot and the computer) and K+1 classes of customers (K subsystem events plus control events represented by displayed 4-D errors in manual control mode) was developed.

To obtain the stationary policy (i.e., to determine the values of M and m) suitable for the experimental situation, a computer simulation was performed. Poisson arrivals and Erlang service time distributions for subsystems were assumed. The K subsystem tasks were preempted by the control task whenever it occurred. The system was represented as a preemptive resume priority queueing system: $(M/E_k/2):(PRP/K/K)$ with implemented threshold control.

A simple case was considered in which the model parameters were determined in the following manner. 1) Subsystem arrival rates, service rates, and waiting cost rates were all uniform among the subsystems. Furthermore, $c_1 = c_2 = \ldots = c_K = 1$ was used. 2) Two levels of arrival rates were assumed, i.e., low arrival (at 0.0167 events per second per subsystem) and high arrival (at 0.0333 events per second per subsystem). 3) Pilot performance in terms of service rates, service errors and control services were obtained from the experiment discussed in the next section. 4) The computer aiding employed the same service rates as the pilot and automatically went off when no event needed service (i.e., m=0).

The results based on the computer simulation of 10,000 events for K=6 and desired server occupancy for the pilot of $\rho = 0.7$ showed that, without control task, M=7 for low arrival and 3 for high arrival; with control task, M=3 for low and 1 for high arrival. This choice of 0.7 as the desired server occupancy is based on the observation of simple queueing systems where a higher value of occupancy will result in a steep rise in queue length and wide fluctuations in traffic flow.

The values obtained above are the threshold values which the computer should employ to adapt to both the subsystem arrival rate and the control task involvement to minimize expected subsystem waiting time subject to the desired occupancy level. For systems with different values of K, λ, μ, or ρ etc., the appropriate threshold values are likely to be different from those listed above. These values could be determined using the computer simulation with the parameters modified appropriately.

Prediction of system performance by the model was also obtained through the computer simulation. The results will be discussed in a later section.

3.4 Experimental Design

An experiment based on the representation described above was conducted. First, four subjects, all very knowledgeable of the system, were used in a preliminary experiment. Another eight trained subjects, all of them male students in engineering, then participated in a balanced sequence of sixteen experimental runs (see Table 3-1) with different workload levels. This was achieved by combining three levels of control task involvement (perfect autopilot, manual control, autopilot with possible malfunctions), three levels of subsystem event arrival rates (no arrival, low arrival, high arrival), and three levels of availability of computer aiding (no aiding, aiding with fixed switching policy, and aiding with adaptive policy). For each experimental run, the subject was first told the specific tasks to perform, then a 14-minute trial was given, and a questionare (in the form that is shown in Appendix V) was filled out by the subject.

For the experimental runs with perfect autopilot, only the subsystem task was considered. An "autopilot" kept the aircraft on course and on schedule. These runs served as baseline performance for the subsystem task. In the manual control runs, the subject had to perform both subsystem and control task. He was told that the control task was more important than the subsystem task. For the runs where autopilot maufunctions were possible, the autopilot was available during most of the experiment such that the subject was not required to fly the airplane except to occasionally check autopilot performance. As soon as he detected an autopilot malfunction, which was characterized

	Subject 1 5	Subject 2 6	Subject 3 7	Subject 4 8
Autopilot without Malfunction	(training)	(training)	(training)	(training)
	low arrival with aiding	low arrival without aiding	high arrival with aiding	high arrival without aiding
	low arrival without aiding	low arrival with aiding	high arrival without aiding	high arrival with aiding
	high arrival with aiding	high arrival without aiding	low arrival with aiding	low arrival without aiding
	high arrival without aiding	high arrival with aiding	low arrival without aiding	low arrival with aiding
Manual Control	(training)	(training)	(training)	(training)
	no arrival	no arrival	no arrival	no arrival
	low arrival with aiding	low arrival without aiding	high arrival with aiding	high arrival without aiding
	low arrival without aiding	low arrival with aiding	high arrival without aiding	high arrival with aiding
	high arrival with aiding	high arrival without aiding	low arrival with aiding	low arrival without aiding
	high arrival without aiding	high arrival with aiding	low arrival without aiding	low arrival with aiding
Autopilot with Malfunction	(training)	(training)	(training)	(training)
	no arrival	no arrival	no arrival	no arrival
	low arrival with aiding	low arrival without aiding	high arrival with aiding	high arrival without aiding
	low arrival without aiding	low arrival with aiding	high arrival without aiding	high arrival with aiding
	high arrival with aiding	high arrival without aiding	low arrival with aiding	low arrival without aiding
	high arrival without aiding	high arrival with aiding	low arrival without aiding	low arrival with aiding
	low arrival adaptive aid	low arrival adaptive aid	high arrival adaptive aid	high arrival adaptive aid
	high arrival adaptive aid	high arrival adaptive aid	low arrival adaptive aid	low arrival adaptive aid

Table 3-1. Design of experiment.

by the airplane deviating from the mapped course at a rate of one degree per second, he was required to take over the flight control task, and fly the airplane back to the mapped course. In this case, the airplane would lock on the desired course as soon as it flew within a slowly-expanding circle around the on-schedule circle, and the autopilot mode was restored. The autopilot malfunction happened relatively infrequently, based on a Poisson distribution with mean inter-arrival time of 160 seconds.

After the pilot detected the autopilot malfunction, he had to devote a major portion of his attention to the control task, leaving subsystem tasks less attended and thus, risk and uncertainties grew as subsystem event detection and service were further delayed. This is one of many situations in which airborne computer aiding is more valuable. Also, in this period, the pilot's workload suddenly increased. To adapt to this type of change, a lower threshold value can be used to reduce subsystem service delay and pilot workload.

Based on this idea, two experiment runs with adaptive computer aiding were included in the set of runs with autopilot malfunctions possible. Instead of using M=3 all the time as in the fixed threshold policy, the adaptive policy used M=1 whenever the pilot was in manual mode. In total, there were seven experimental runs with autopilot malfunction: one run with no subsystem arrival (serving as a baseline performance for malfunctions), two runs with no aiding, two with fixed-threshold aiding, and two with adaptive aiding. This arrangement allowed for the evaluation for the effectiveness of computer aiding and further the benefit of the adaptive policy beyond that of fixed aiding.

Three or more, depending on the task situation, of the following performance measures were evaluated in every experimental run:

1. average delay in response and service for subsystem events,

2. subsystem service errors (e.g., false alarms, incorrect actions, etc.),

3. 4-D RMS and average flight course errors (distance, schedule, and altitude errors),

4. flight control inputs including aileron, elevator, speed, etc.,

5. detection and service times for autopilot malfunctions,

6. server occupancy in terms of the fraction of time the subject was performing either subsystem or control tasks,

7. subjective ratings of level of effort required for the tasks and the desirability of computer aiding.

All these measures were obtained by analyzing the sampled data. The subsystem event response time was measured from the time of event occurrence to the time at which an action was initiated. The service time was measured from the time of last action initiation to the time of action completion for the event. The waiting time was measured from the time of event occurrence to the time of action completion for the event. Waiting time is equal to the sum of response time and service time only when the event is serviced by one server and no incorrect action occurs. The empirical results along with the analyses of variance are discussed in the next chapter.

4. RESULTS

The results presented in this chapter include experimental results obtained from the flight management situation (section 4.1), the results from the simulation program, and the comparison of the two sets of results (section 4.2).

4.1 Experimental Results

The data sampled during the flight management experiment was analyzed to obtain the seven objective measures listed in the previous chapter. The subjective ratings of the task situations based on the questionnaire answered by the subjects during the experiment were also obtained. For each of these measures, factors of significance were determined using the analysis of variance and the underlying trends of variation are investigated. Finally, a correlation test was conducted between subjective effort rating and the measured server occupancy.

4.1.1 Objective measures

An analysis of variance was conducted for each performance measure. (ANOVA tables appear in Appendix VI.) Effects were accepted as significant if $p \leq 0.05$. For the mean subsystem waiting time averaged across the subjects

(Figure 4-1), all three experimental variables (i.e., the level of control involvement, the level of subsystem event arrivals, and the level of availability of computer aiding) produced statistically significant effects. The hypothesis that the mean waiting times at the three levels of control involvement (i.e., autopilot, manual, and autopilot malfunction modes) are all equal was rejected. Similar results were obtained for the two levels (low or high) of subsystem arrival rates and the two levels (with or without) of availability of computer aiding (Table VI-1). Thus, as shown in Figure 4-1, the subsystem waiting time increased as the subsystem arrival rate increased, as the control involvement increased, and as the aiding availability decreased. The interaction between control mode and aiding type are also found to be significant (Table VI-1). However, the effect was not substantial compared to the main effects (Figure 4-1). A separate test showed that the adaptive policy also produced significant improvement (Table VI-2). The adaptive aiding further reduced the subsystem waiting time beyond the fixed-threshold aiding, even though the adaptive policy was only utilized during a small portion of the total task time (i.e., only when there were malfunctions).

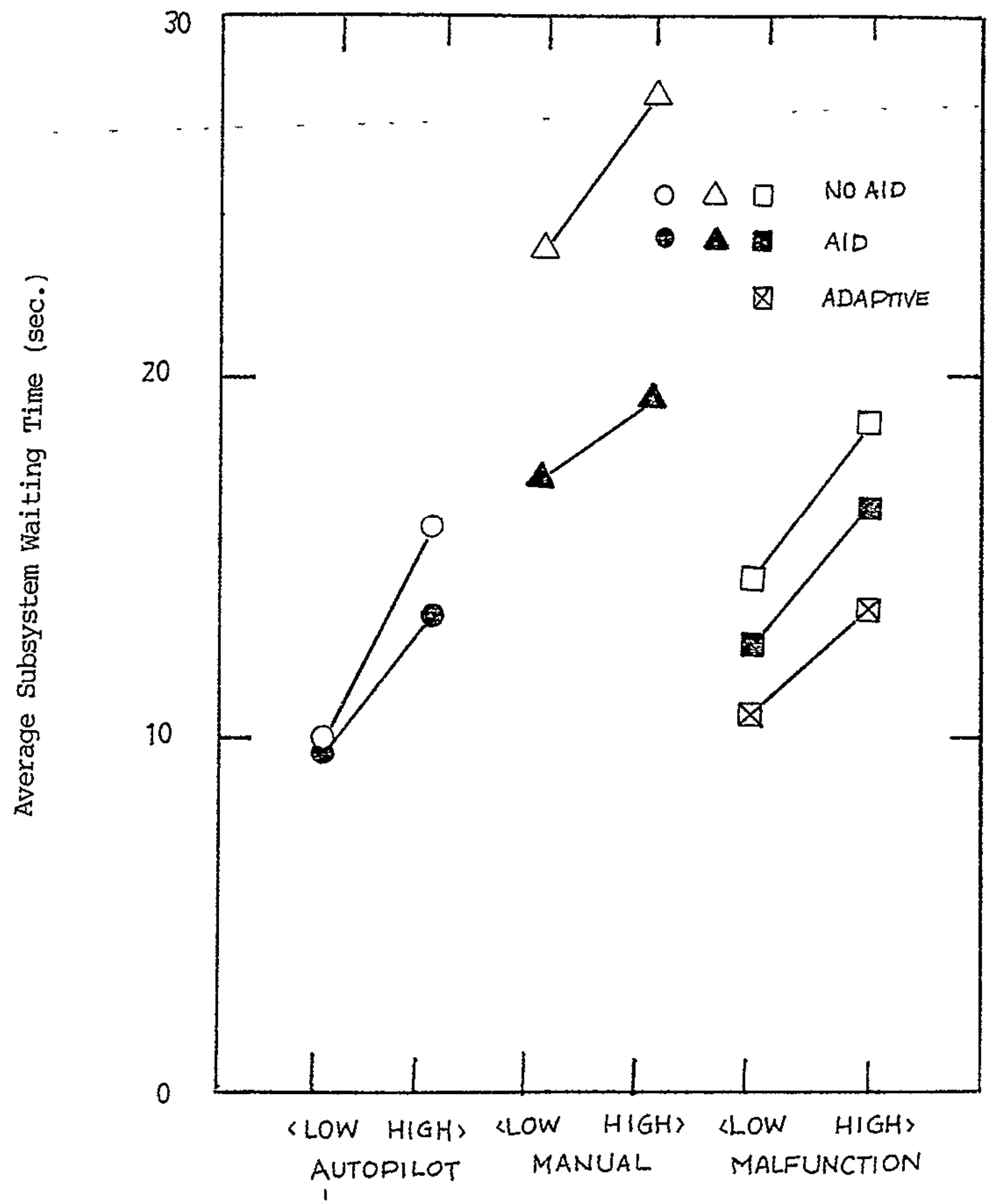

Figure 4-1. Average subsystem waiting time.

The average subsystem service time, shown in Table 4-1, appears to be independent of subsystem event arrival rate and the availability of computer aiding. The measured service time increased as control involvement increased, due to the preemption of subsystem service by the control tasks. It is reasonable here to assume that the subsystem service time obtained in the autopilot mode may serve as the baseline estimate of the service time the subjects had to devote to the subsystems.

Service errors (false alarms and incorrect actions) were counted. The false alarm arrival rates (Figure 4-2) were then calculated as the inverse of the average inter-arrival time of false alarms (i.e., as the mean frequency of false alarms during the server idle period). The probabilities of incorrect actions (Figure 4-3) were calculated as the ratio of the number of incorrect actions to the total number of actions. Both false alarm arrival rate and probability of incorrect action increased as subsystem arrival rate increased. Thus, with the arrival levels used in the experiment, higher subsystem arrival caused a deviation of human workload from optimal in terms of increased service errors. On the other hand, a lower arrival level (of Ø.Ø111 arrivals per second) used in the previous experiment had shown that lower arrival rates would also cuase a deviation from optimal workload. This evidence

Arrival rate	Aiding type	Subsystem service time Mean (sec.)	Variance (sec.^{2})
Autopilot mode			
Low	No aiding	5.56	0.49
Low	Aiding	5.60	0.58
High	No aiding	5.78	0.48
High	Aiding	5.74	0.59
Manual mode			
Low	No aiding	8.54	33.16
Low	Aiding	7.44	14.76
High	No aiding	7.59	18.68
High	Aiding	7.69	13.47
Malfunction mode			
Low	No aiding	6.16	10.93
Low	Aiding	6.14	11.58
Low	Adaptive aiding	5.96	21.77
High	No aiding	6.27	19.87
High	Aiding	6.33	36.29
High	Adaptive aiding	5.86	13.76

Table 4-1. Subsystem service time.

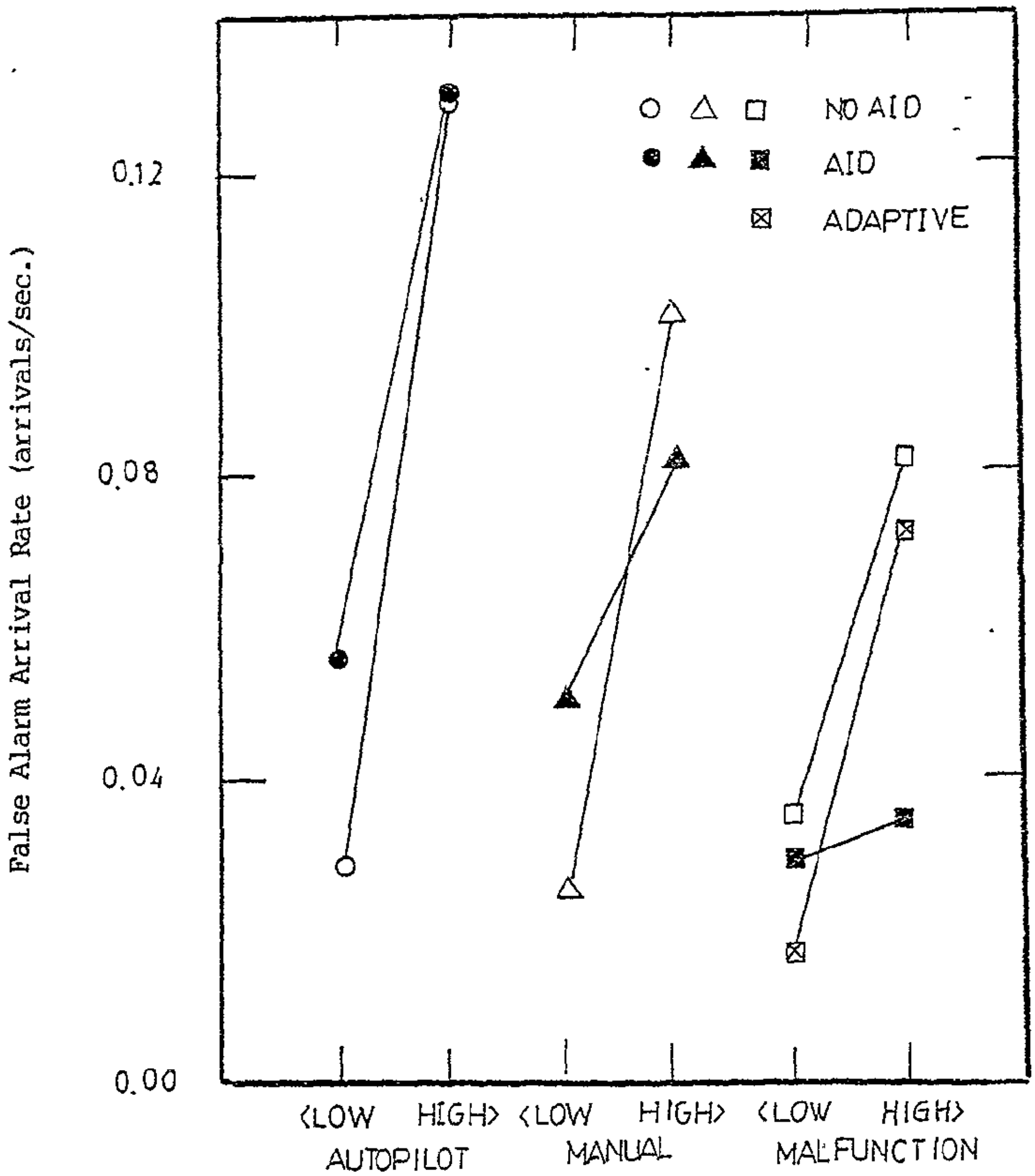

Figure 4-2. False alarm arrival rates.

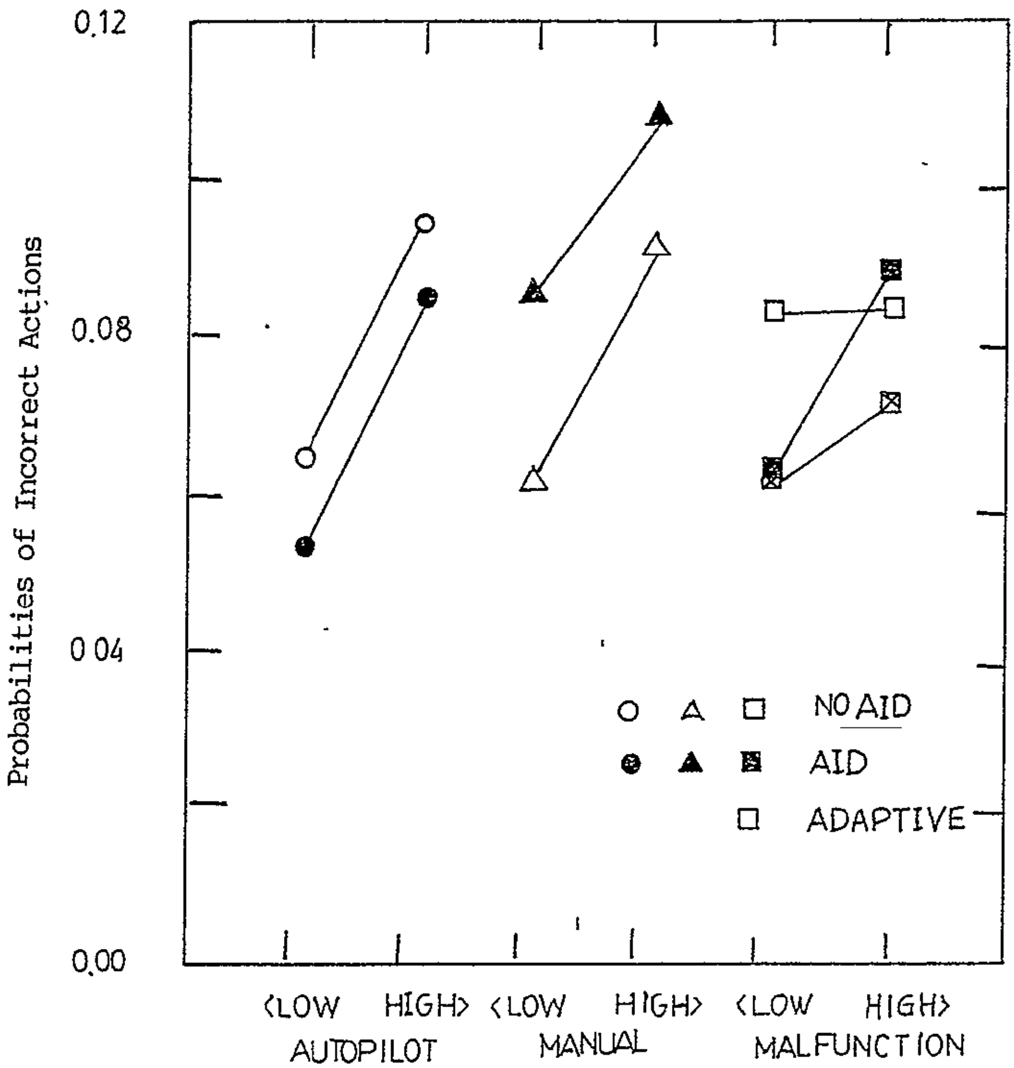

Figure 4-3. Probabilities of incorrect actions.

supports the notion of the existence of an acceptable workload range for the human in decision making tasks.

The control task performance in terms of the mean and RMS schedule errors, mean and RMS altitude errors, and mean distance errors is shown in Table 4-2. To show the effect of the experiment variables on control errors, the RMS distance error is presented in Figure 4-4. The analysis of the RMS distance error indicated that both the level of control involvement (Tables VI-3 and VI-4) and the mere presence of subsystem tasks (Table VI-5) significantly affected the control error. No consistent variation in this distance error was observed as the subsystem arrival rate or aiding situation varied. The lower RMS distance error for the autopilot malfunction mode probably resulted from the subjects' more intense attention to the control task in the case of autopilot malfunction.

The RMS values of roll and pitch angles across the subjects were obtained and are shown in Figure 4-5 and 4-6. Only control mode had a significant effect (Tables VI-6, -7, -8) on the RMS roll angle. The subjects were observed to control more frequently and to use more extreme control actions to fulfill the malfunction task requirements than when in the normal manual mode.

Arrival rate	Aiding type	Schedule error Mean (ft.)	Schedule error RMS (ft.)	Altitute error Mean (ft.)	Altitute error RMS (ft.)	Mean distance error (ft.)
Manual mode						
No	No aiding	1969	2405	854	3852	1523
Low	No aiding	2677	3569	98	4059	2155
Low	Aiding	2675	3216	1189	4758	2029
High	No aiding	3580	4818	245	5794	1842
High	Aiding	2832	3567	895	4298	1708
Malfunction mode						
No	No aiding	1079	1487	364	858	961
Low	No aiding	1647	2752	581	1736	1521
Low	Aiding	1431	2292	777	2391	1333
Low	Adaptive aiding	2265	3413	2905	6100	1844
High	No aiding	1650	2455	1716	4916	1480
High	Aiding	2548	3929	2022	5382	2023
High	Adaptive aiding	1388	2017	550	1788	1255

Table 4-2. Control task performance.

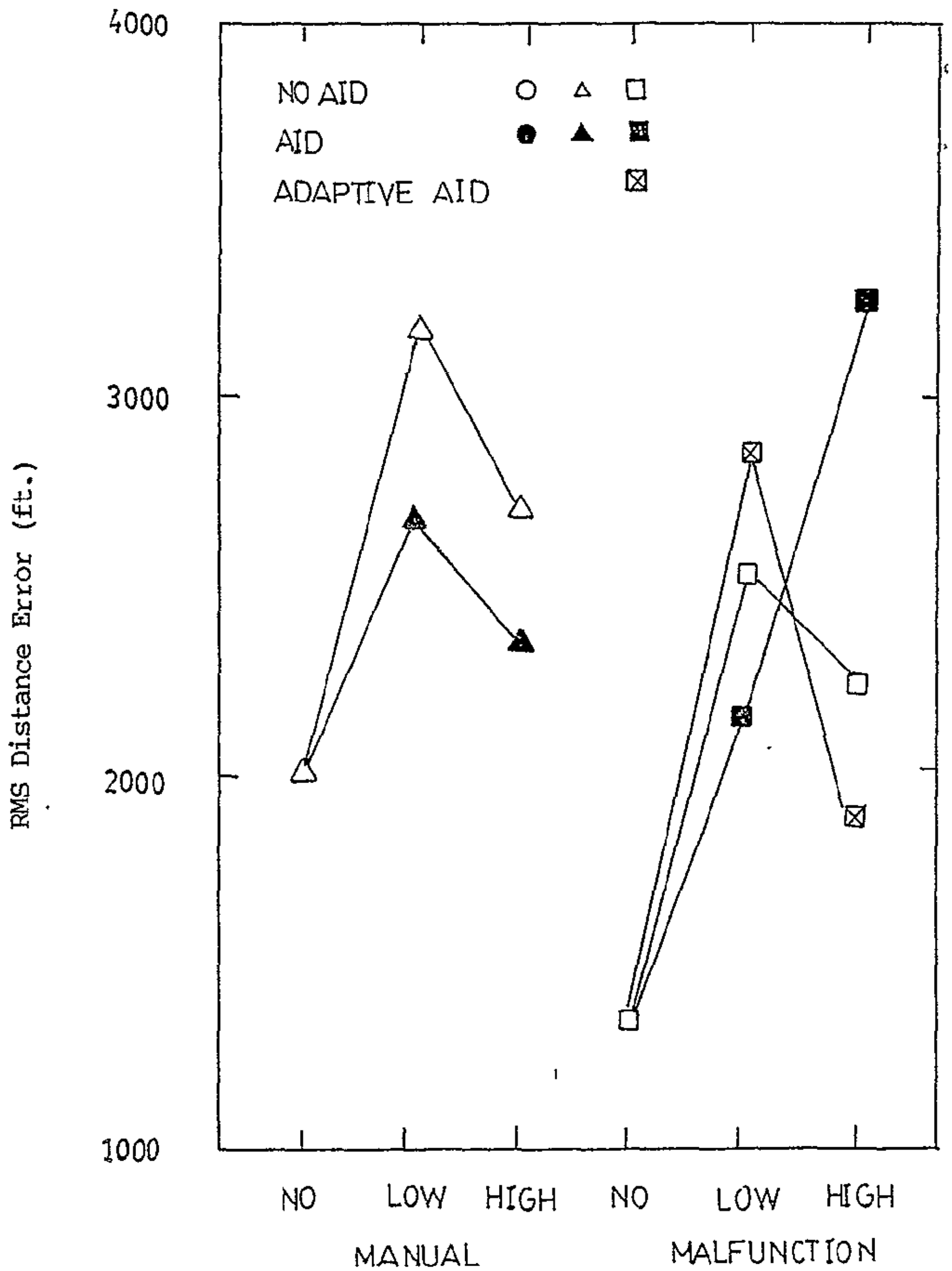

Figure 4-4. RMS distance errors.

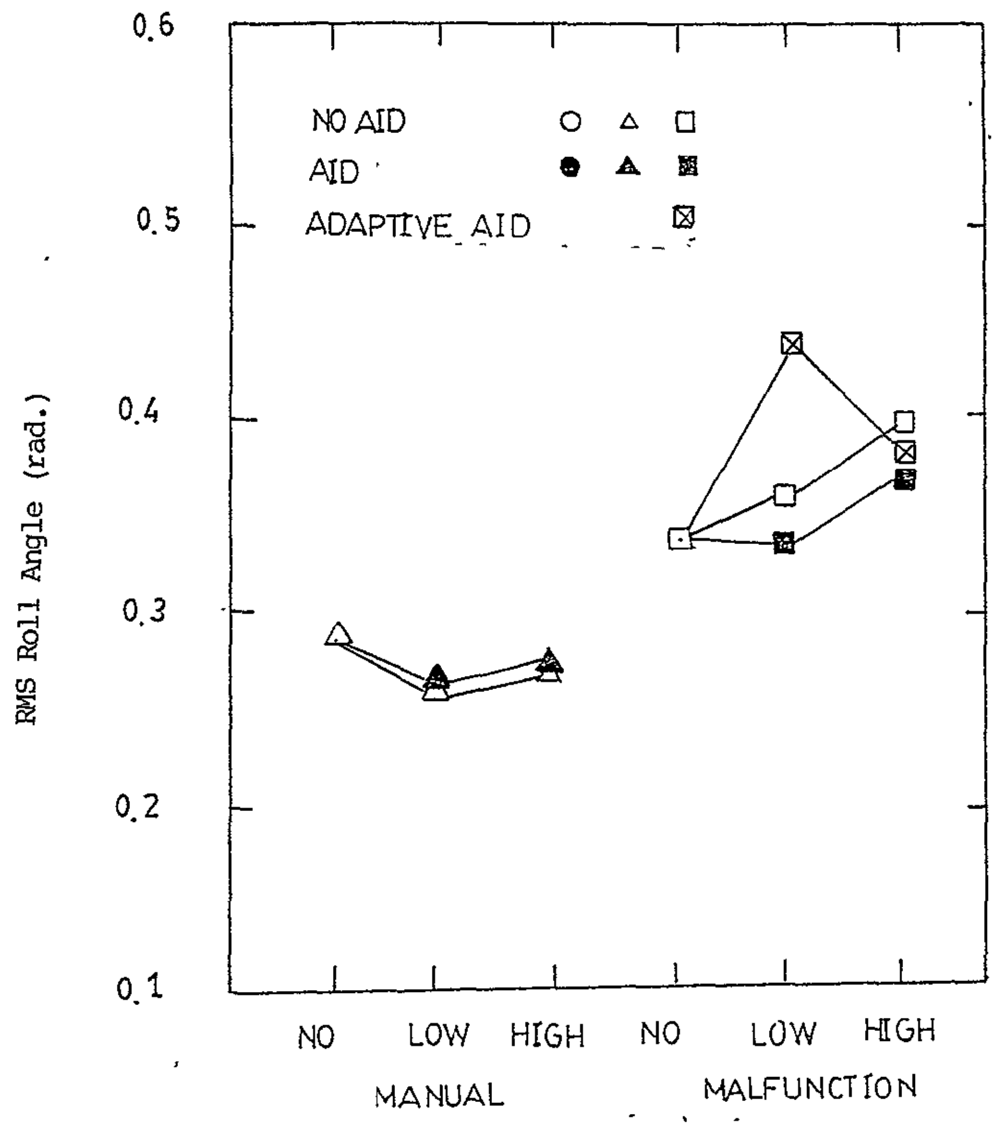

Figure 4-5. RMS roll angles.

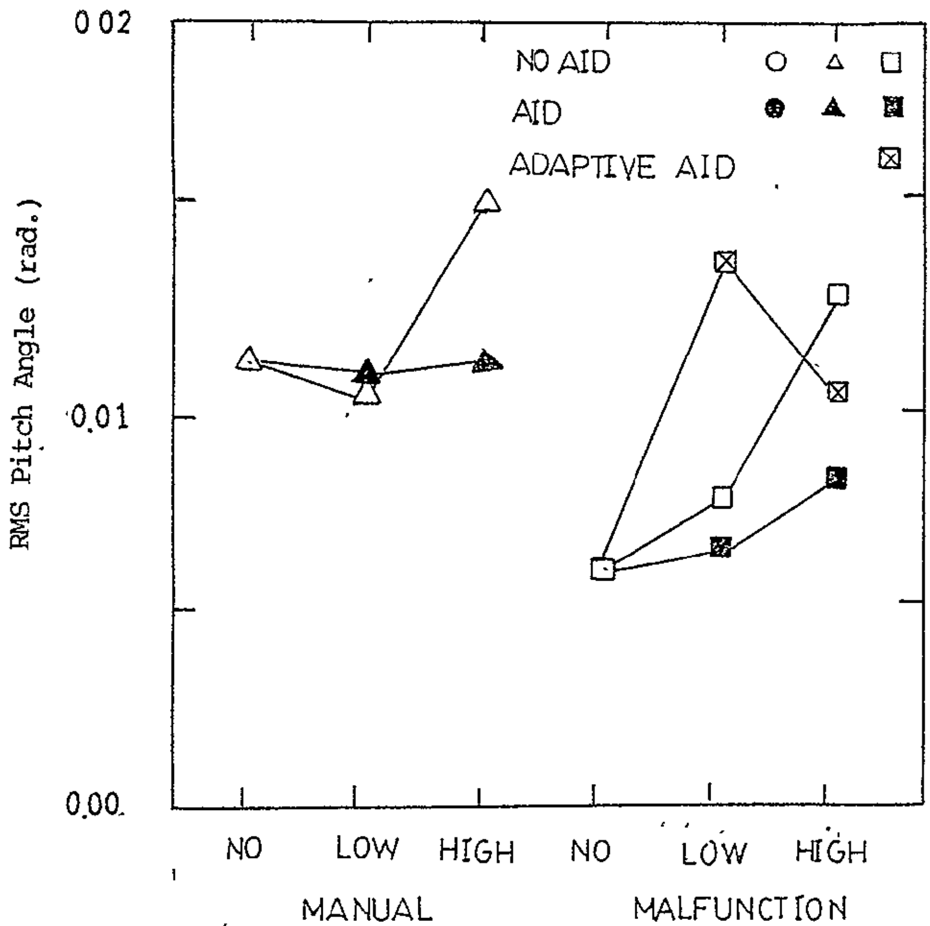

Figure 4-6. RMS pitch angles.

To illustrate typical control stick input samples, Figure 4-7 plots the sampled aileron stick input versus time for two experimental runs (one in the manual and one in the autopilot malfunction mode). It can be seen that discrete control actions were adopted by the subjects and the intensity of control effort may be appropriately measured by the frequency and duration of the control action. It appears that a skillful subject would wait until some of the observed errors (e.g., schedule/distance errors, altitude error, etc.) exceeded certain thresholds and then a control action was promptly initiated.

For the aileron control, the control duration (measured as the period from initiation until release of the control sticks) and arrival rate (measured as the inverse of average stick idle time) were averaged across subjects and are shown in Table 4-3. Elevator control by itself was more stable and demanded less attention, and when it was employed to adjust the altitude, it usually required a 'bang-off-bang' type attention. It is thus assumed that a maximum of 1.5 seconds each was spent by the subjects in the beginning and the end of the elevtor control. The combined aileron plus elevator control durations and arrival times were then calculated, and are listed in Table 4-3. These values served as input to the simulation program to be discussed in the next section. Again, the level of control involvement

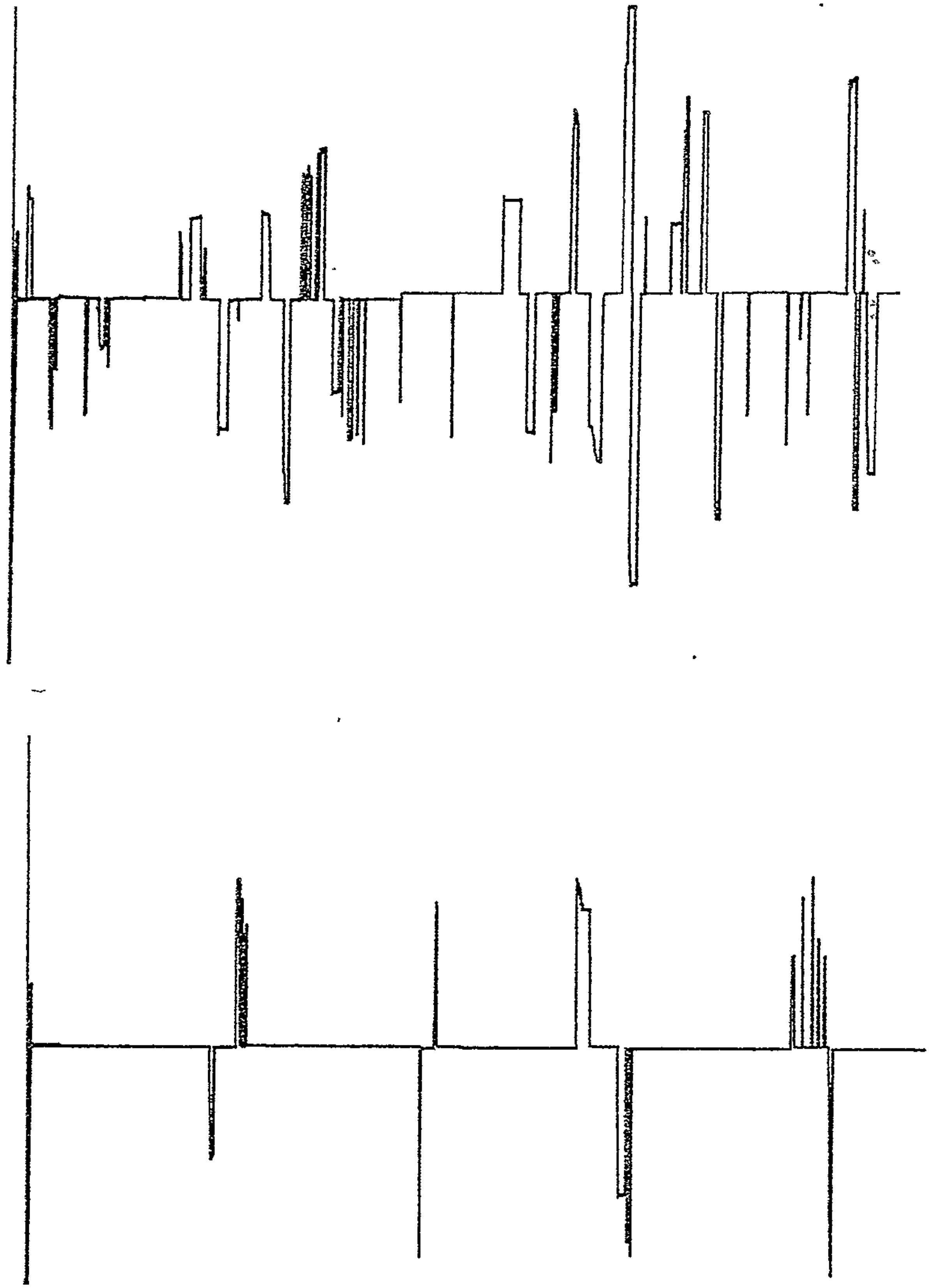

Figure 4-7. Aileron control input in manual mode (top) and in malfunction mode (bottom).

Arrival rate	Aiding type	Aileron control Duration (sec.)	Aileron control Arrival (sec.$^{-1}$)	Aileron+elevator control Duration (sec.)	Aileron+elevator control Arrival (sec.$^{-1}$)
Manual mode					
No	No aiding	2.48	0.13	2.37	0.28
Low	No aiding	1.73	0.09	1.90	0.22
Low	Aiding	1.97	0.09	2.04	0.20
High	No aiding	2.02	0.08	2.10	0.16
High	Aiding	2.34	0.08	2.25	0.18
Malfunction mode					
No	No aiding	3.52	0.17	2.99	0.41
Low	No aiding	3.00	0.15	2.69	0.32
Low	Aiding	3.23	0.16	2.97	0.32
Low	Adaptive aiding	3.65	0.15	3.32	0.28
High	No aiding	3.23	0.16	2.93	0.33
High	Aiding	3.27	0.14	2.97	0.28
High	Adaptive aiding	3.25	0.16	3.04	0.29

Table 4-3. Mean duration and mean arrival rate of control actions.

was significant. It appears that the subjects, in general, employed longer and more frequent control action in malfunction situations than in the normal manual mode and also in the situations without subsystem tasks as compared to those with subsystem tasks.

The server occupancies averaged across subjects in the various task situations were calculated (using the control parameters estimated earlier whenever the control task was involved), and are presented in Figure 4-8. As expected, all three experiment variables were significant in affecting this measure (Table VI-9). The adaptive policy seems to reduce the server occupancy further, however, the effect is not significant (Table VI-10).

The service time and detection time for autopilot malfunction were measured and are shown in Table 4-4. The analysis showed that no consistent variation of these measures with respect to the experimental variables was observed and no effect of statistical significance was obtained.

4.1.2 Subjective ratings

Subject's ratings concerning the perceived level of effort in performing the tasks, the effectiveness, and the desirability of computer aiding, and the ease of interaction

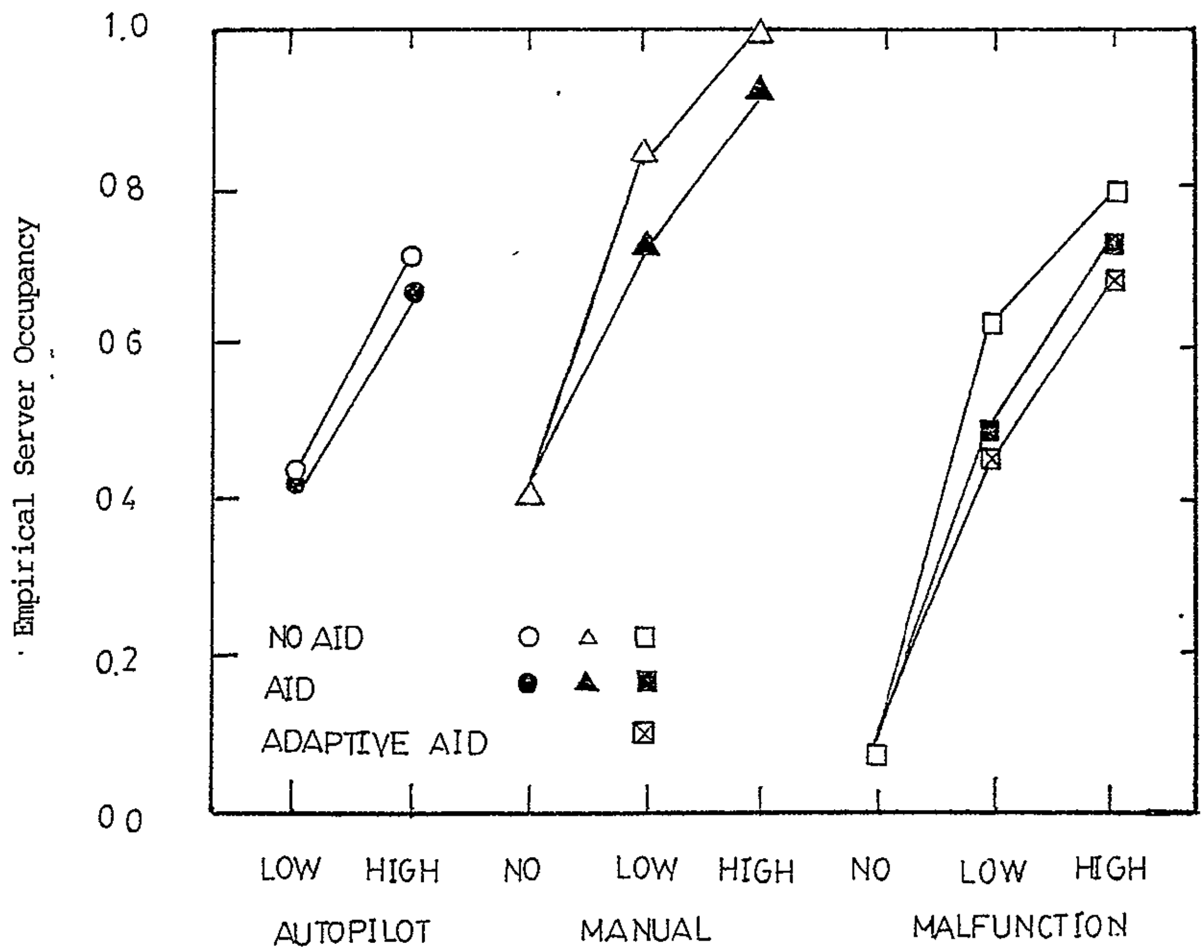

Figure 4-8. Empirical server occupancy

Arrival rate	Aiding type	Detection time Mean (sec.)	Detection time Variance (sec^2)	Service time Mean (sec.)	Service time Variance (sec^2)
No	No aiding	6.69	4.91	23.27	260.35
Low	No aiding	7.04	7.80	27.09	284.25
Low	Aiding	7.22	8.99	29.07	423 34
Low	Adaptive aiding	7.17	3.94	33.35	1240.48
High	No aiding	7.44	10.19	32.83	885.38
High	Aiding	8.43	11.14	36.16	1922.17
High	Adaptive aiding	6.95	5.72	29.27	578.89

Table 4-4. Detection time and service time of autopilot malfunction.

with the aiding were analyzed. Individual ratings for different task situations were first converted to a normalized scale, then these measures of variation among tasks were averaged across the subjects. The resulting effort ratings (Figure 4-9) were shown to be affected by all the experiment variables, which included level of aiding availability, level of control involvement, interaction of aiding and level of control, and subsystem event arrival rate (Table VI-12). The perceived level of effort increased as control involvement increased, as subsystem arrival increased, and as computer availability decreased. The effect of adaptive computer aiding was not found to be significant. This is probably because the adaptive aiding used did not lead to a significantly lower overall server occupancy, and also because the adaptive policy was employed rather infrequently. Further, when it was being used, the subjects usually were too involved with restoring the autopilot to notice the fact that the computer was helping more often than usual.

The subjective ratings of the various aspects of computer aiding appear to vary less among the subjects than those of effort ratings. The aiding was considered 'easy to interact with' (Figure 4-10) and 'desirable' by the subjects (Figure 4-11). Its effect on performance improvement was perceived to be from 'slight' to 'large' (Figure 4-12).

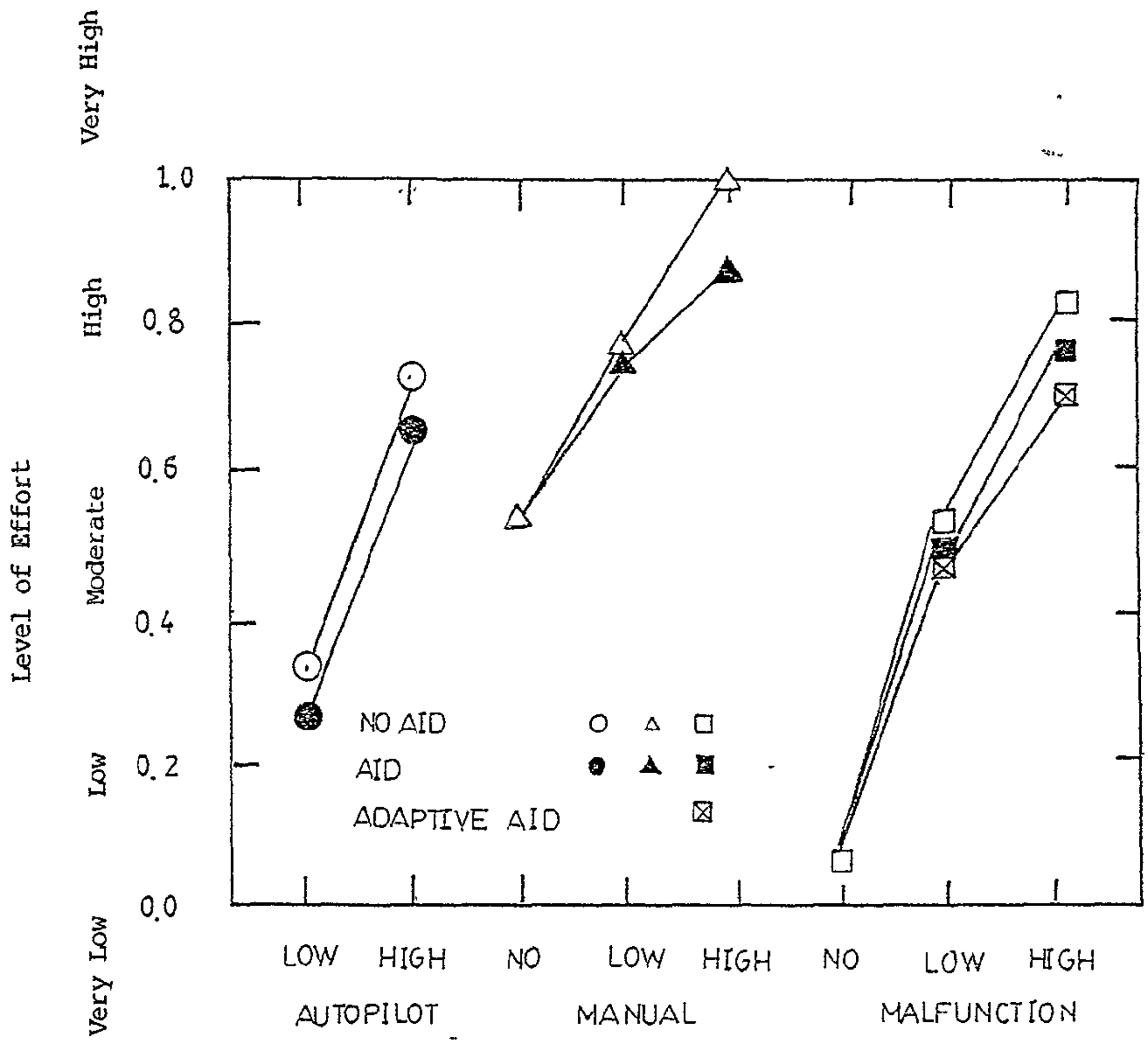

Figure 4-9. Subjective ratings of effort.

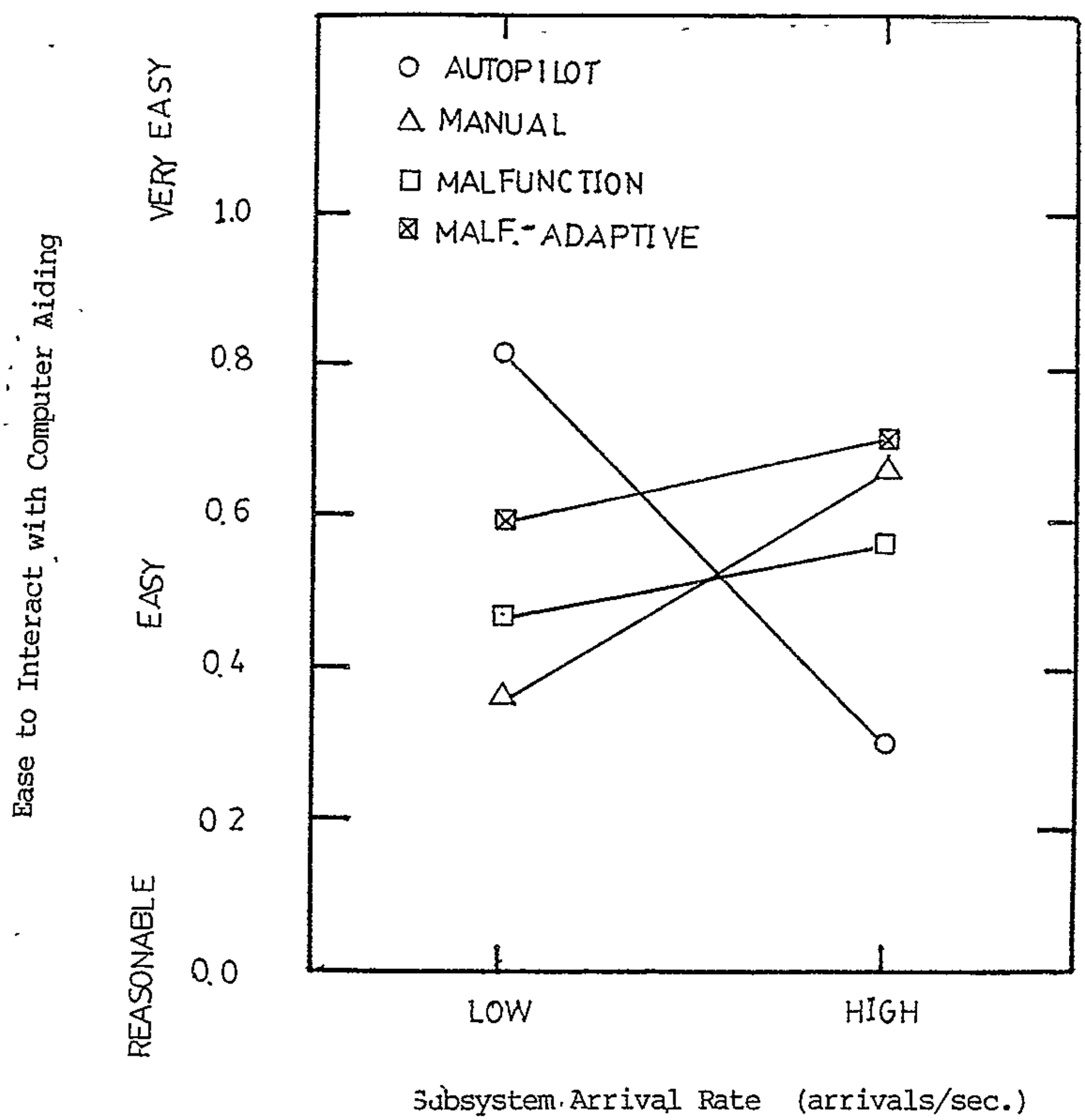

Figure 4-10. Subjective ratings of the ease of interaction with computer aiding.

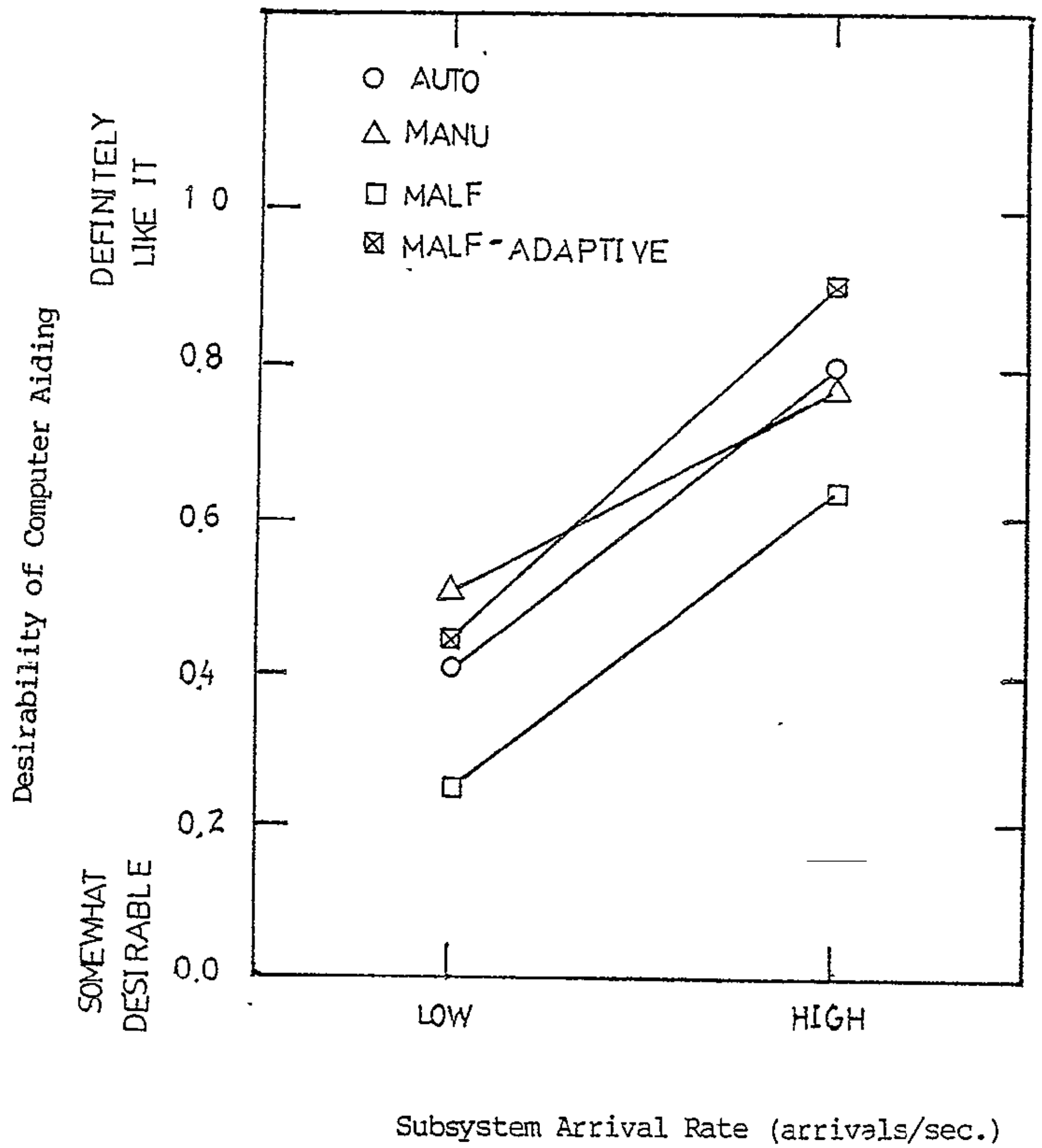

Figure 4-11. Subjective ratings of the desirability of computer aiding.

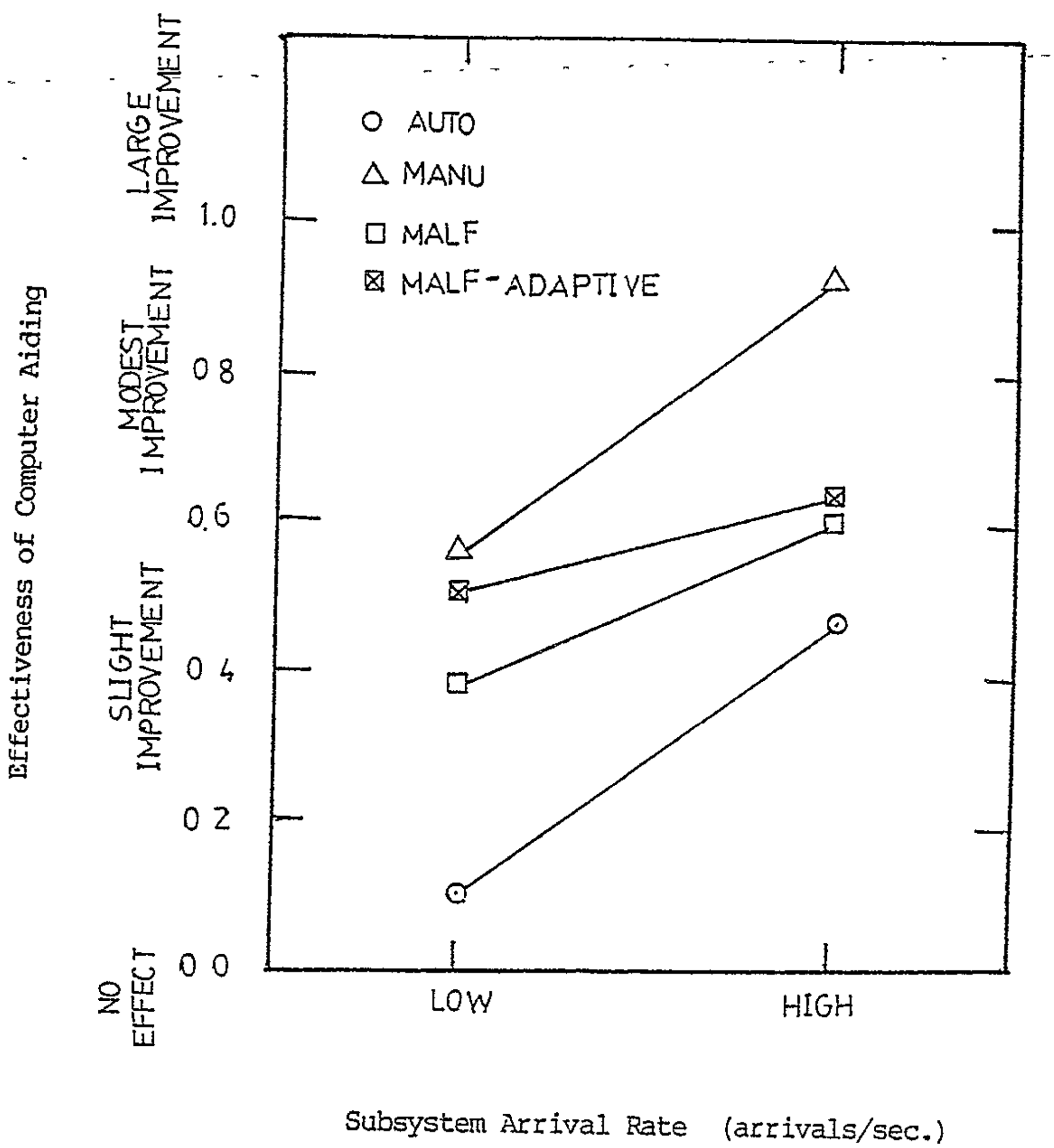

Figure 4-12. Subjective ratings of the effectiveness of computer aiding.

Among the factors of significance the subjects saw the aiding to be relatively more effective and more desirable in the situations of high arrival rates, and in the situations with high control involvement (Tables VI-14, -15, -16, -17). It is interesting to note that even though the subjects did not fully perceive the effectiveness of adaptive aiding beyond a fixed-threshold aiding, they did confirm the desirability of adaptive aiding. As far as the ease of interaction is concerned, variations in the main effects did not consistently affect subjects' perceptions of the ease with which they could interact with the computer aiding. There is, however, a strong interaction between the effects of control mode and subsystem arrival rate in affecting this measure. A possible reason is that, in the autopilot mode, the higher subsystem arrival rate increased the likelihood of the subjects's noticing the computer's request for possible preemption. While in the manual and malfunction modes, where the subjects were involved in the control tasks, they probably did not notice as much the computer requesting possible preemption. Thus, we can conjecture that the form of preemption designed into the system discussed here may require more thought in terms of ease of interaction.

In general, based on the comments from the subjects, it appears that all subjects were quite in favor of both the aiding scheme used in the experimental situation and the general computer aiding idea.

4.1.3 The correlation of subjective effort ratings and server occupancy

One particularly important issue in research into pilot workload is the development of a technique for reliable prediction of the effort the pilot exerts to meet predetermined levels of input load [Smit, 1976]. Among the common workload measures are physiological measurements, task performance, and subjective ratings. In the multi-task situation, the performance in terms of server occupancy provides not only a direct measure of fraction of time the pilot is busy but also an indirect estimate of the intensity of attention that the tasks demand of the pilot. Thus, the measure of server occupancy seems appropriate to serve as pilot workload indicator in a multi-task flight management situation.

While the measurement of server occupancy and its relationship to average queue length and hence intensity of demand is somewhat of a conjecture and deserves further exploration, an accurate correlation of this measure to the subjective effort rating may offer promise for this approach. The empirical occupancy data is plotted versus the subjective effort ratings in Figure 4-13. The correlation coefficient r was computed and found to be 0.950. The hypothesis that the two measures are uncorrelated was also tested using a student-t test. It was rejected at 0.0025 level (t_{14} = 11.35).

4.2 Simulation Results and Comparison

The simulation approach proposed in Chapter Two, incorporated with the flight management specifications in Chapter Three, provides a model representation reasonably close to that of the flight management experiment. A detailed program flow diagram is shown in Appendix VII.

Human false alarms, human control actions and autopilot malfunctions were considered to be separate processes with given arrival and service statistics and with appropriate interactions with each other. Features of computer aiding such as the preemption period were easily implemented. To provide comparable results with those from the experiment, the following parameters were specified for the program using values corresponding to those in the experiment:

1. subsystem arrival rates, (0.0167, and 0.0333 events per second for low and high arrival, respectively),
2. subsystem scanning time, (0.25 seconds per subsystem for human, 0.0 for computer),
3. monitor/control attention shift (0.2 seconds),

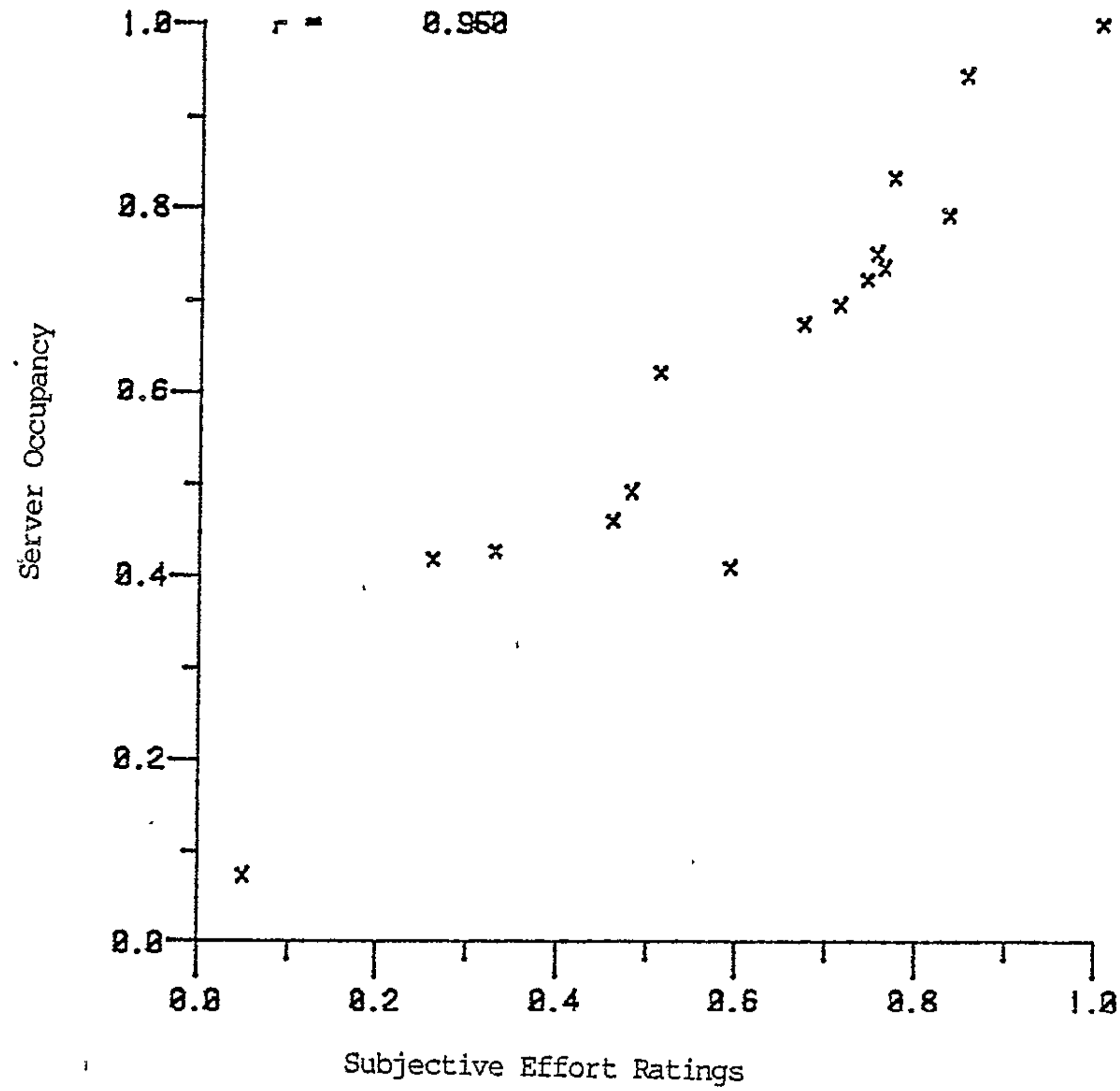

Figure 4-13. Correlation of empirical server occupancy and subjective effort ratings.

4. computer service time (7 seconds),
5. autopilot malfunction arrival rate (0.0667 arrivals per second).

In the simulation program, all process arrivals (including subsystem arrivals, false alarm arrivals, autopilot malfunction arrivals, and control action arrivals) were generated using a Poisson distribution, and all service times (including service of subsystem events, incorrect actions, false alarms, autopilot malfunctions, and control actions) were according to Erlang-k distribution. In the cases of subsystem and false alarm services, the service time distributions were approximately constant. The set of variables used in the program represent values measured from the experiment and averaged across all appropriate situations. These variables served as input to the program and included:

1. the subsystem service time distribution (with mean of 5.668 seconds and k = 62),
2. the control service time distribution (with mean of 2.13 seconds for manual mode, and 2.99 seconds for malfunction mode, k = 2 for both),
3. the false alarm arrival rates (with mean of 0.00344 arrivals per second for low arrival rate, and 0.00915 arrivals per second for high arrival rate),
4. the probabilities of incorrect actions (of 0.0656 for low arrival rate, and of 0.0865 for high arrival rate),
5. the autopilot malfunction detection and service time distribution (with mean of 7.28 seconds and 30.15 seconds respectively, and k = 2 for both),

C-2

6. the incorrect action service time distribution (with mean of 3.50 seconds and k = 5),

7. false alarm service time distribution (of constant 1.56 seconds)

Subsystem waiting time and server occupancy statistics are shown in Table 4-5. All parameters in the model were either predetermined or empirically measured and no adjustments were made. A comparison of the average subsystem waiting time and server occupancy with those measured from the experiment is shown in Figures 4-14 and 4-15. A statistical test was conducted for both measures of all experimental cases. The hypothesis that the model results and the empirical data have the same set of mean values was not rejected at the 5% significance level.

The variance of the waiting time is relatively high in some cases, resulting from a saturation of arrivals which caused subjects to be in a very high workload situation. Other than that, the model's predictions are very good, especially for average waiting time in autopilot mode and for server occupancy in autopilot malfunction mode. In addition, a high correlation (r=0.96) is found between this model occupancy and the subjective effort ratings.

A better understanding of the control task mechanism and a better estimate of control task parameters will further improve the model accuracy. In this respect,

Arrival rate	Aiding type	Subsystem waiting time (sec.) Mean	Standard deviation	Server occupancy
Autopilot mode				
Low	No aiding	9.89	5.07	0.424
Low	Aiding	9.70	4.09	0.424
High	No aiding	14.01	11.68	0.725
High	Aiding	12.15	6.35	0.684
Manual mode				
No	No aiding	--	--	0.406
Low	No aiding	18.47	14.04	0.727
Low	Aiding	16.50	8.74	0.715
High	No aiding	30.12	36.89	0.900
High	Aiding	18.20	10.32	0.853
Malfunction mode				
No	No aiding	--	--	0.095
Low	No aiding	11.88	7.70	0.509
Low	Aiding	11.69	6.68	0.501
Low	Adaptive aiding	10.54	4.68	0.478
High	No aiding	17.30	16.99	0.770
High	Aiding	13.71	7.98	0.735
High	Adaptive aiding	12.69	6.67	0.718

Table 4-5. Subsystem waiting time and server occupancy from simulation model.

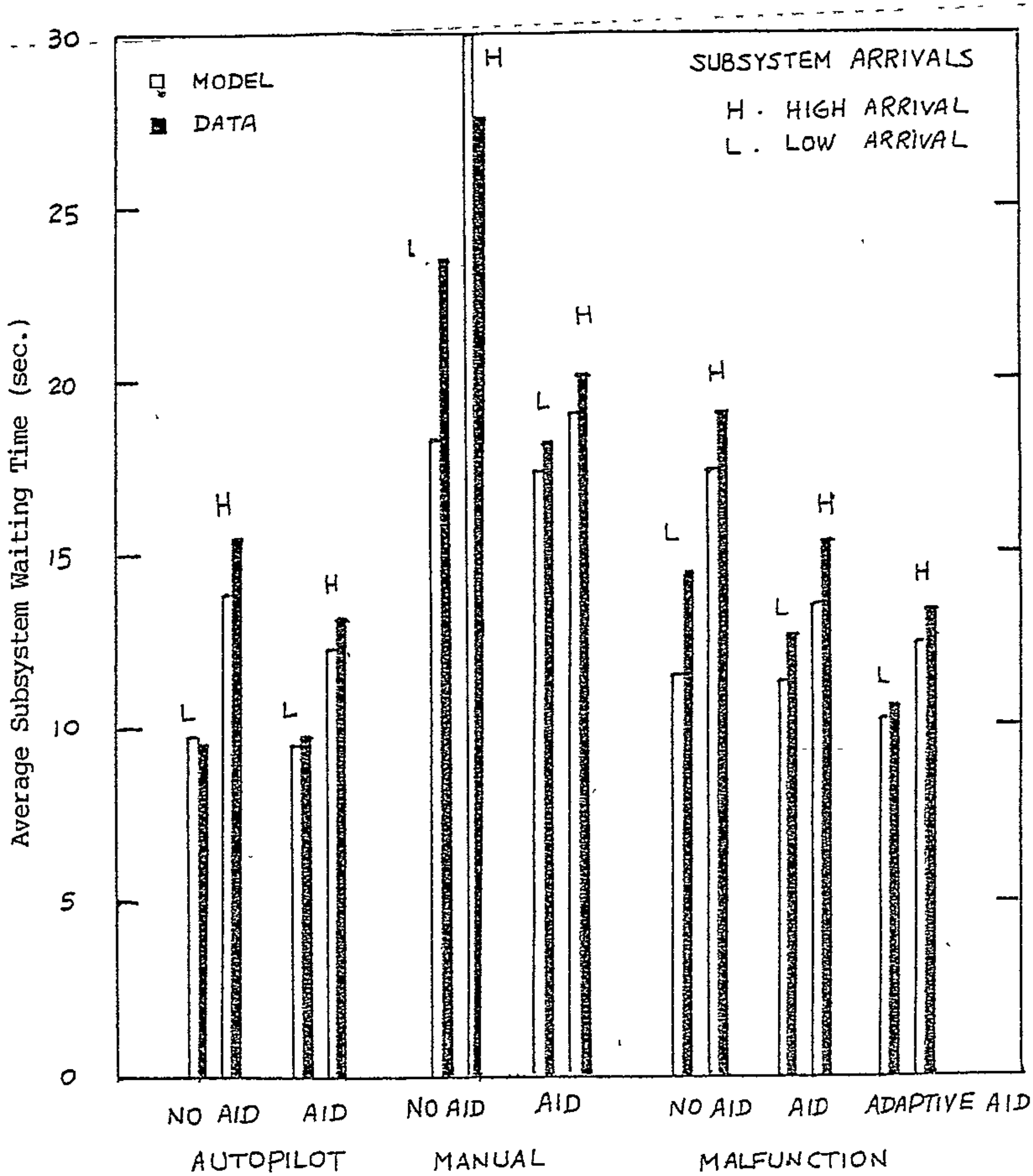

Figure 4-14. Comparison of empirical and model average subsystem waiting time.

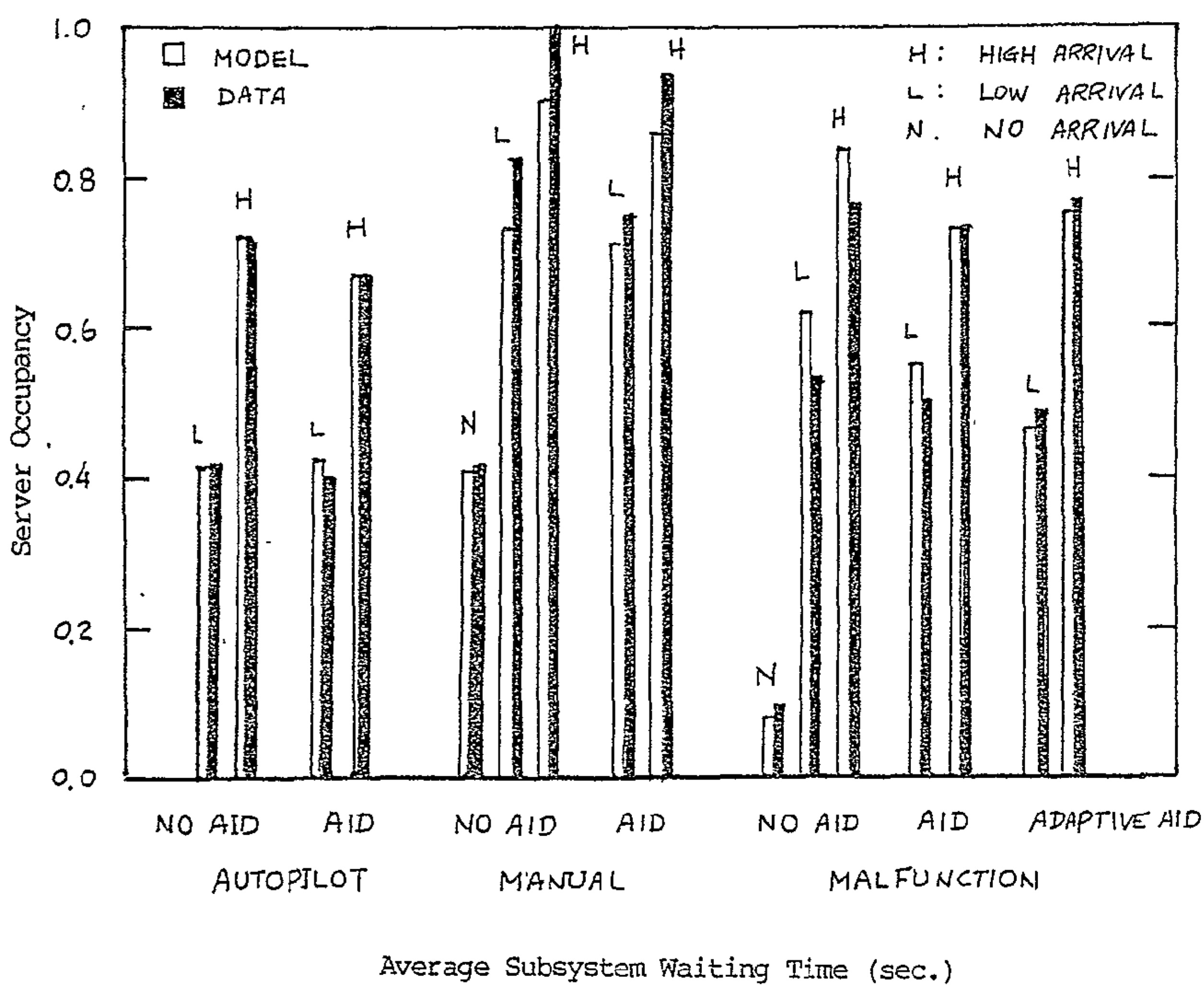

Figure 4-15. Comparison of empirical and model server occupancy measure.

Govindaraj and Rouse [1978] are currently working on modeling of the human as a controller in multi-task monitoring and control situations.

4.3 Summary

The experimental results (summarized in Table 4-6) show that all the experimental variables were statistically significant in terms of affecting the subsystem waiting time, the server occupancy, and subjective effort ratings. It was observed that systems that are designed to relax control requirements, such as the autopilot, seem to improve both control and subsystem performance, while systems that are designed to relax subsystem requirements, such as computer aiding in monitoring or highly reliable subsystems seem to improve only subsystem performance. The possible reason for this is that the control tasks preempt subsystem tasks, and thus control task inefficiency is likely to affect the performance of subsystem tasks; the reverse in not true.

Server occupancy and subjective effort ratings were highly correlated. Aiding enhanced system performance in terms of subsystem average waiting time, server occupancy and subjective effort ratings. Adaptive aiding was shown to further reduce subsystem waiting time. Interestingly, adaptive aiding did not significantly affect subjective

	Arrival rate	Control mode	Aiding	Adaptive aiding
Subsystem waiting	⇑ *	⇑	⇓	⇓
Effort ratings	⇑	⇑	⇓	-
Server occupancy	⇑	⇑	⇓	-
Distance error	⇑	⇑	-	-
Roll angle	-	⇓	-	-
Aiding effectiveness	⇑	⇑		-
Aiding desirability	⇑	⇑		⇑

* ⇑ - significant increase
⇓ - significant decrease
\- - not significant.

Table 4-6. Summary of average main effects.

effort ratings, and it did not significantly improve service occupancy. However, it did improve system performance. Consequently, the subjective effort ratings are more closely tied to service occupancy than to system performance.

The queueing model fits the experiment results reasonably well, especially when one considers that no parameter adjustments were made. A better representation of control task preemption is needed to further improve model accuracy in predicting system performance. Although the parameters used are based on averages across the subjects, it is simple to adjust the parameters according to each individual and thus, the model is adaptable to individual differences.

5. DISCUSSION AND CONCLUSIONS

The purpose of this research has been to present a unified formulation of computer-aided multi-task decision making and to develop a general strategy for allocation of decision making responsibility between human and computer. An experimental study was conducted in the context of a flight management situation. A model based on a queueing theory framework appears adequate to represent the situation and flexible enough for future implementation. The allocation policy also seemed to be well-accepted by the subjects in the experiment. Implementation issues and model applications are to be discussed in the following sections.

5.1 Implementation Issues

The simulation of an airborne flight management system became a major research tool for validating the human decision making model and investigating the proposed allocation policy. The development of a more detailed task scenario to represent a real-world flight management situation seems necessary before the proposed scheme is actually tested and implemented on a full-scale simulator or real aircraft. Issues involving representation of the pilot's task and measurement of parameters are of particular importance in this respect.

5.1.1 Task representation

The aiding system would be most beneficial if it was designed to be of use during the whole flight mission from take-off until landing. A multiple mode scenario generator including take-off, climb-out, enroute, approach, and landing modes may seem desirable and feasible. In the enroute mode, which is the setting of the current experimental situation, the pilot should be allowed to plan flight paths by inserting way points, editing a route and estmating the speed and time to each points, etc. and then the automated navigation system would assure that these specifications were met. In different flight operation modes, the requirements and priorities are different. An information display similar to the Master Monitor Display reported by Hughes Aircraft Co.[1974] seems appropriate to handle this multi-mode operation. Information concerning system states and status as well as alerts and warnings are centralized in an integrated display. Pilot information seeking behavior becomes the focus of research within this approach [Rouse and Neubauer, 1978].

Considering the subsystem tasks, the checklist structure may be expanded to meet flight operation requirements such as those described in the DC-10 Flight Crew Operating Manual [Douglas Aircraft Co., 1975]. Tasks may be organized either by functional classification (e.g., divided into subsystems such as power plant, landing gear, pneumatic, navigation, fuel, electrical, communication, flight control, hydraulic systems, etc), or by procedural classification (e.g., emergency, abnormal, normal, and conditional procedures, etc.). The functional subsystems are composed of several sublevels of physical component units, and are under pilot's supervision. Abnormal signals in the low level units are assumed to propagate upward to the top level to signify the occurrance of an event, while a top-down tree-structured checklist could guide a procedure for diagnosis purposes. Considerations of importance in fault diagnosis situations are discussed by Rouse [1978].

The aircraft dynamics, on the other hand, can simply be modified or reprogrammed to represent the specific type of aircraft of interest. The autopilot malfunctions could occur in a variety of ways. In addition to the current presentation of a random deviation from the prescribed map course, other modes of malfunction include deviation from the normal speed setting or a loss of altitude.

With the above considerations in mind, it seems that implementation in an aircraft simulator would provide a reasonable level of reality.* Furthermore, this simulator

* In fact, such implementation in a GAT-2 simulator is planned as the next phase of this research.

could provide not only the necessary motion and alerting cues (audio, symbolic cues, etc) but also extra room and speed (because of the availability of analog devices for the aircraft dynamics and instruments) for the use of digital computation. This would enhance the realization of the allocation algorithm and the design of a realistic task scenario. In addition, after the new experimental situation is established, regular pilots may be used as the subjects.

5.1.2 Task dependency

Because the sublevel units of the subsystems may be inter-connected physically, the subsystem arrivals may not be independently distributed as assumed in the previous analysis. As a result, a specific subsystem event arrival distribution may depend on the overall time-variant subsystem configuration: $(n_1, n_2, \ldots, n_K;\tau)$, where n_k is the number of events in subsystem k and τ represents the mode or phase of system operation. With this change, the problem certainly becomes more complicated. However, the approach in Chapter Two is still applicable. The proposed algorithm could be followed using

$$\lambda = \sum_{\kappa=1}^{K} \lambda_\kappa (n_1, n_2, \ldots, n_K;\tau),$$

and then employing a threshold M as a function of arrival rate λ. Hopefully, only a small number of tasks would be inter-dependent, and the space needed to store these functional relations would not be too large. Another problem that arises is the estimation of the set of parameters $\lambda_K(n_1, n_2, \ldots, n_K; \tau)$, which is to be discussed in the next section.

5.1.3 Parameter estimation and measurement

The parameters used in the system, mainly the arrival rates, service rates, and cost rates, may be obtained by pre-determined statistics or by on-line measurement. The arrival and service information of subsystems may be pre-determined by an analysis of field data. For example, the event arrival information used in an air traffic controller model [Schmidt, 1978] was based on reports of Couluris, et al. [1974] and of Hunter, et al. [1974]. Since the pilot or the operator would also perform information checking, adjustment, and mode selection etc., in addition to fault correction; the normal reliability data of the subsystems are usually not adequate, and data collected from field observation would be necessary. The process for collecting these data probably is not more complicated or difficult than the regular reliability and maintainability analysis for the subsystems.

On-line measurement of these data may be facilitated by using real-time human-to-computer communication channels, mentioned in Section 1.4, such as the physiological EEG measure and statistical model matching. The use of scalp recorded, cortical event related potentials (ERP) [Wickens, et al.,1977] seems to successfully predict the human's detection and reaction to an event. Thus, the statistics of event arrivals perceived by the human can be easily collected and used as the simulation input.

The priority assignment and the waiting costs of subsystem events may also depend on system configuration: $(n_1, n_2, \ldots, n_K;\tau)$. For example, the functioning of landing gear requires more attention during the landing phase than it requires in the enroute mode. Even in the same operational mode, the rapid increase of risk and uncertainty due to delay in servicing some subsystems may result in a change in relative priories. A time-dependent priority, other than the fixed priority rule that was used in the flight management experimental situation, may be appropriate. Kleinrock's [1976] priority queue model may be applicable in this case. This approach assumes a proportionality between priority of subsystem i, $p_i(t)$, and its waiting time: $p_i(t) = b_i\,(t-t_i)$, where b_i is a constant and t_i is the time of event arrival in the subsystem i.

The relative waiting costs of subsystems, if they are to be measured on-line, may be represented as negative utility functions which may be assessed by using a dynamic utility estimation technique [Freedy, et al., 1976]. This approach, combined with a pattern recognition technique developed by Freedy and his colleagues, might enable the computer decision maker to learn the human's utility function by watching his activities. This would also be useful in avoiding conflicts.

As discussed in the last chapter, the server occupancy measure appears to be a good workload index which can both predict system performance (e.g., effort ratings and subsystem waiting time), and be responsive to changes in task demands (i.e., the subsystem arrival rate). As with the previous parameters, both field estimation and on-line measurement are applicable to this parameter. However, this measure is usually evaluated in stationary processes, and an on-line evaluation requires a continuous updating of the ratio of busy to total time over a given time duration. Besides, due to possible preemption among tasks, error in this measure could accumulate unless every instance of human initiation and relinquishing of action can be accurately detected and recorded (so as to determine when a busy period starts and when it ends). A moving average estimate combined with the ERP measure seems to solve these problems.

Given this, the question of what is an acceptable level of workload, in terms of human occupancy, will be easier to answer.

5.2 Extensions of the Approach

It would be rather simple for the model to be adapted to new task implementations. Particularly, because each subsytem unit is simulated as an individual process flow, the program is modularly expandable as far as the interaction (activation/deactivation, preemption, etc.) among processes is concerned. Time-varying priorities would be easy to implement in the program. In general, the queueing framework lends the model flexibility to incorporate a number of find-grained models emphasizing signal detection, attention allocation, information processing, or utility assessment aspects of human decision making.

The adaptive policy discussed in Chapter Two is only partially realized and verified in the experiment. Due to limited task variations, the allocation policy was only used to adapt to autopilot malfunctions. Even in this situation, the adaptive policy proposed is seen only in some cases to realize an advantage from the additional information of malfunction occurrence. Therefore the policy is only qualified as "quasi-adaptive" [Bertsekas, 1976]. The adaptive features of the policy would stand out better in less structured problems. However, experimental justification of using this policy in a truly dynamic sense would prove expensive at this stage. Therefore, we have been satisfied with a scheme of setting up a stationary policy and performing on-line estimation and table look-up. There certainly are other schemes which would be superior to this open-loop feedback type scheme [Bertsekas, 1976]. They are, however, much more complicated.

5.3 Applications

The approach espoused in this thesis is applicable to many multi-task situations where system criteria and goals are rather clear, computer decision aids are desirable, the tasks to be performed are well-structured, and the time delay of discrete events rather than the deviation of continuous states is of major concern. Situations falling into this category include: flight management, air traffic control, and various industrial process monitoring and control tasks. The design of computer aiding for each of these situations would involve developing an experimental situation, conducting experiments, measuring parameters, and analyzing the cost-effectiveness of the predicted performance improvements. The procedure and example in this thesis may also serve as a guide line for the design of multi-task decision making systems in those other situations.

REFERENCES

1. Aoki, M. and Toda, M. (1975) "Parameter adaptive resource allocation problem for a decentralized dynamic system," IEEE Trans. Automat. Contr., Vol. AC-20, No. 2, Apr. 1975, pp. 192-200.

2. ARMA (1977) Operating Manual of Automated Navigation System for British Airways, ARMA, Division of AMBAC, Garden City, New York, Oct. 1977.

3. Bagchi, T.P. and Cunningham, A.M. (1972) "Bayesian approach to the design of queueing systems," INFOR 10-1, pp. 36-46.

4. Balachandran, K.B. (1973) "Control policies for a single service system," Management Science, 19-9, pp.1013-18.

5. Balachandran, K.B. and Tijms, H. (1975) "On the D-policy for the M/G/1 queue," management Science, 21-9, pp.1073-6.

6. Bar-Shalom Y., Larson, R.E., and Grossberg, M.A. (1974) "Application of stochastic control theory to resource allocation under uncertainty," IEEE Trans. Automat. Contr., Vol. AC-19, No. 1, Feb. 1974, pp. 1-7.

7. Bell, C.E. (1971) "Characterization and computation of optimal policies for operating an M/G/1 queueing system with removable server," Oper. Res., 19-1, pp. 208-18.

8. Bell, C.E. (1973) "Optimal operation of an M/G/1 priority queue with removable server," Oper. Res., 21-6, pp. 1281-90.

9. Bertsekas, D.P. (1976) Dynamic Programming and Stochastic Control, New York: Academic Press.

10. Blakelock, J.H. (1965) Automatic Control of Aircraft and Missiles, New York: John Wiley Sons, Inc.

11. Carbonell, J.R. (1966) "A queueing model of many-instrument visual sampling," IEEE Trans. Human Factors in Electronics, Vol. HFE-7, Dec. 1966, pp. 157-64.

12. Carbonell, J.R. (1968) "A queueing model of many-instrument visual sampling, experimental validation," IEEE Trans. Man-Machine systems, Vol. MMS-9, No.3, Sep.1968, pp. 82-7.

13. Chong, C-Y and Athans, M. (1971) "On the stochastic control of linear systems with different information sets," IEEE Trans. Automat. Contr., Vol. AC-16, No. 5, Oct. 1971, pp. 423-30.

14. Chu, Y. (1976) "Optimal control of queues," Seminars on Queueing Systems, ed. Rouse, W. B., University of Illinois, UILU-ENG-77-4002, Dec. 1976.

15. Chu, Y. (1977) "Optimal adaptive allocation of decision making responsibility between human and computer in multi-task situations," in Pilot Interaction with Automated Airborne Decision Making Systems, Semiannual Progress Report, NASA-NSG-2119, May 1977, pp. 5-42.

16. Chu, Y. and Rouse, W.B. (1977) "Optimal adaptive allocation of decision making responsibility between human and computer in multi-task situations," Proceedings of the International Conference on Cybernetics and Society, pp. 168-175, September 1977.

17. Connelly, M.E. (1977) Simulation Studies of Airborne Traffic Situation Display Applications, Report for FAA office of Systems Engineering Management, ESL-R-751, MIT, May 1977

18. Couluris, G.J., et al. (1974) Capacity and Productivity Implications of Enroute ATC Automation, Stanford Research Institute for DOT/FAA, Report FAA-RD-74-196, Dec. 1974.

19. Cox, D.R. and Smith, W.L. (1961) Queues, London: Methuen.

20. Douglas Aircraft Company (1975) DC-10 Flight Crew Operating Manual, System Description; Reference and Performance, Long Beach, Calif.

21. Duff, I.S. (1977) "A survey of sparse matrix research," Proc. of the IEEE, vol. 65, No. 4, April 1977, pp. 500-535.

22. Enstrom, K.D. and Rouse, W.B. (1977) "Real-time determination of how a human has allocated his attention between control and monitoring tasks," IEEE trans. Syst., Man. Cybern., Vol. SMC-7, No. 3, Mar. 1977.

23. Etkin, B. (1972) Dynamics of Atmospheric Flight, New York: Wiley.

24. Findeisen, W., Brdys, M., Tatjewski, P., Malinowski, K., and Wozniak, A. (1978) "On-line hierarchical control for steady-state systems," IEEE Trans. Automat. Contr., Vol. AC-23, No. 2, Apr. 1978, pp. 298-305.

25. Firschein, O., Fischler, M.A., Coles, L.S., and Tenenbaum, J.M. (1974) "Intelligent machine are on the way," IEEE Spectrum, Vol. 11, No. 7, pp. 41-38, July 1974.

26. Forestier, J.P., and Varaiya, P. (1978) "Multilayer control of large Markov chains," IEEE Trans. Automat. Contr., Vol. AC-23, No. 2, Apr. 1978, pp.189-209.

27. Freedy, A., Davis, K.B., Steeb, R., Samet, M.G., and Gardiner, P.C. (1976) Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation and Application, Perceptronics Report PFTR-1016-76-8/30, Aug. 1976.

28. Gopinath, B., and Morrison, J.A. (1977) "Discrete-time single server queues with correlated inputs," The Bell System Technical Journal, Vol. 56, No. 9, Nov. 1977, pp. 1743-68.

29. Govindaraj, T. and Rouse, W.B. (1978) "Modeling the human as a controller in a multitask environment," Proceedings of the Fourteenth Annual Conference on Manual Control, University of Southern California, April 1978.

30. Greenstein, J.S. and Rouse, W.B. (1978) "A model of event detection in multiple process monitoring situations," Proceedings of the Fourteenth Annual Conference on Manual Control, University of Southern California, April 1978.

31. Gregory, R.L. (1966) Eye and Brain: the Psychology of Seeing. New York: McGraw-Hill.

32. Heyman, D.P. (1968) "Optimal operating policy for M/G/1 queueing systems," Operations Research, 16, pp. 362-82.

33. Heyman, D.P. (1977) "The T-policy for the M/G/1 queue," Management Science, 23-7, March 1977, pp. 775-8.

34. Ho, Y-C and Chu, K-C (1972) "Team decision theory and information structures in optimal control problems-Part I and Part II," IEEE Trans. Automat. Contr., Vol. AC-17, No. 1, Feb. 1972, pp. 15-28.

35. Hsuan, J. and Shaw, L. (1976) "Optimal control and dynamic repairman assignment for a linear stochastic system," ORSA/TIMS, March 1976, Philadelphia.

36. Hughes Aircraft Company, Aerospace Group (1974) Master Monitor Display Study, Naval Air Devel. Center, Report.

37. Hunter, J.S., Blumenfeld, D.E., and Hsu, D.A. (1974) "Modeling Air Traffic Performance Measures, Vol.1, Message Element Analysis and Dictionaries, DOT/FAA Report FAA-RD-73-147, July 1974.

38. Jacobus, C.J. (1975) M and M System Design and Operation, University of Illinois, Coordinated Science Laboratory Report R-709, Dec. 1975.

39. James, E.B. and Partridge (1972) "Machine intelligence: the best of both worlds?" Int. J. Man-Mach. Stud., Vol. 4, No. 1, pp. 23-31.

40. Johannsen G. and Rouse W.B. (1978) "Prospects of a mathematical theory of human behavior in complex man-machine system tasks," Proceedings of the Fourteenth Annual Conference on Manual Control, University of Southern California, April 1978.

41. Kleinrock, L. (1976) Queueing Systems, Vol.2: Computer Applications, New York: John Wiley Sons.

42. Licklider, J.C.R. (1960) "Man-computer symbiosis," IEEE Trans. on Human Factors in Electron., Vol. HFE-1, No. 1, pp. 4-11, March 1960.

43. Man, F.T. (1973) "Optimal control of time varying queue," Management Science, 19-11, pp.1249-56.

44. Martin, J. (1973) Design of man-computer dialogues. Englewood Cliffs, NJ: Prentice-Hall, 1973.

45. Miller, G.A. (1974) "Needed: a better theory of cognitive organization," IEEE Trans. System, Man, Cybern., Vol. SMC-4, Jan. 1974, pp. 95-7.

46. Mesarovic, M.D., Macko, D., and Takahara, Y. (1970) Theory of Hierarchical, Multilevel, Systems, New york: Academic Press.

47. Moray, N., ed. (1978) Mental Workload, New York: Plenum Press.

48. Mova, V.V. and Ponomarenko, L.A. (1974) "On the optimal assignment of priorities depending on the state of a servicing system with a finite number of waiting places," Trans. Engr. Cybernetics, Vol. 12, No. 5, Sep.-Oct. 1974, pp.66-72.

49. Nazarov, A.A. and Terpugov, A.V. (1976) "Adaptation in controlled queueing systems," Automatic and Remote Control, Dec. 1976, pp. 1039-42.

50. Nazarov, A.A. (1976) "Optimal formation of queues in a multichannel queueing system," Automatic and Remote Control, 1976, pp. 1242-4.

51. Pasmooij, C.K., Opmeer, C.H.J.M, and Hyndman, B.W. (1976) "Workload in air traffic control - a field study," in T.B. Sheridan and G. Johannsen, Eds., Monitoring Behavior and Supervisory Control, New York: Plenum, 1976, pp. 245-259.

52. Pinneo, L.R. (1975) "Persistent EEG patterns associated with overt and covert speech," Bulletin of the Human Factors Society, Vol. 18, No. 2, pp. 1-2, Feb. 1975.

53. Rijnsdorp, J.E. and Rouse, W.B. (1977) "Design of man-machine interfaces in process control," in H.R. Van Nauta lemke and H.B. Verbruggen, Eds: Digital Computer Applications to Process Control, Amsterdam: North-Holland.

54. Ropelewski, R.R. (1978) "Air Inter's A-300 autolandings routine," Aviation Week and Space Tech., pp. 45-57, April 24, 1978.

55. Rouse, W.B. (1975) "Design of man-computer interfaces for on-line interactive systems," Proceedings of the IEEE, Vol. 63, No. 6, June 1975, pp. 847-57.

56. Rouse, W.B., Chu, Y-Y, and Walden, R.S. (1976) "An experimental situation for study of pilot interaction with automated airborne decision making systems," Proceedings of the Twelfth Annual Conference on Manual Control, May 1976, NASA TM X-73, 170, pp. 39-44.

57. Rouse, W.B. (1977) "Human-computer interaction in multi-task situations," IEEE Trans. System, Man and Cybern., Vol. SMC-7, No. 5, May 1977, PP. 384-92.

58. Rouse, W.B. (1978) "Human problem solving performance in a fault diagnosis task," IEEE Trans. System, Man and Cybern., Vol. SMC-8, No.4, Apr. 1978, pp. 258-271.

59. Rouse, W.B. and Neubauer, H.L. (1978) "Issues in the design of management information systems: a comparison of two very different domains," to appear in the Proceedings of the 1978 IEEE International Conference on Cybernetics and Society, Tokyo, Japan, Nov. 1978.

60. Schmidt, D.K. (1978) "A queueing analysis of the air traffic controller's work load," IEEE Trans. System, Man and Cybern., Vol. SMC-8, No. 6, June 1978, pp. 492-498.

61. Senders, J.W. (1964) "The human operators as a monitor and controller of multidegree of freedom systems," IEEE Trans. Hum. Factors Electron., Vol. HFE-5, No. 1, pp. 2-5, Sept. 1964.

62. Senders, J.W. and Posner, M.J.M. (1976) "A queueing model of monitoring and supervisory behavior," in T.B. Sheridan and G. Johannsen, Eds., Monitoring Behavior and Supervisory Control, New York: Plenum, 1976, pp. 245-259.

63. Sheridan, T.B. and Tulga, M.K. (1978) "A model for dynamic allocation of human attention among multiple tasks," Proceedings of the Fourteenth Annual Conference on Manual Control, University of Southern California, Apr. 1978.

64. Shrage, L. (1975) "Random results in scheduling: implication of queueing theory for scheduling," working paper, University of Chicago.

65. Shaw, L. (1972) "On optimal ramp control of traffic jam queues," IEEE Trans. Automat. Contr., Vol. AC-17, No. 5, Oct. 1972, pp. 630-7.

66. Shaw, L. (1976) "Busy period control of queues based on waiting time at arrivals," Oper. Res., 24-3, 1976, pp.543-63.

67. Smallwood, R.D. (1967) "Internal models and the human instrument monitor," IEEE Trans. Human Factors in Electronics, Vol. HFE-8, No. 3, pp. 181-187, Sept. 1967.

68. Smit, J. (1976) "Pilot workload analysis based upon in-flight physiological measurements and task analysis methods," in Monitoring Behavior and Supervisory Control, Eds., T.B. Sheridan and G. Johannsen, New York: Plenum 1976.

69. Smith, B.A. (1978) "Digital system used to cut workload," Aviation Week and Space Tech., pp. 49-53, March 27, 1978.

70. Sworder, D.D. and Kazangey, T. (1972) "Optimal control, repair, and inventory strategies for a linear stochastic system," IEEE Trans. System, Man and Cybern., Vol. SMC-2, No. 3, July 1972, pp. 342-7.

71. Steeb, R., Artof, M., Crooks, W.H., and Weltman, G. (1975) Man/machine Interaction in Adaptive Computer-aided Control: Final Report. Perceptronics Inc., woodland Hills, CA, NR 196-118, December 1975.

72. Verplank, W.L. (1977) Is There an Optimal Workload in Manual Control? Ph.D. Thesis, MIT, August 1977.

73. Walden, R.S. and Rouse, W.B. (1977) "A queueing model of pilot decision making in a multi-task flight management situation," Proceedings of the Thirteenth Annual Conference on Manual Control, MIT, June 1977.

74. Wempe, T.E. (1974) "Flight management - pilot procedures and system interfaces for the 1980-1990's," AIAA paper No. 74-1297, AIAA Life Sciences and System Conference, Nov. 1974.

75. White, J.A., Schmidt, J.W., and Bennett, G.K. (1975) Analysis of Queueing Systems, New York: Academic Press.

76. Wickens, C.D., Isreal, J., and Donchin, E. (1977) "The event related cortical potential as an index of task workload," Proceedings of Annual Meeting of Human Factors Society, San Francisco, Oct. 1977.

Appendix I

An Analytical Approach to Control of the Second Server in an (M/G/2):(GD/K/K) Queue

The question of interest concerns a queueing system where the second server can be turned on or off so as to optimize system performance with respect to some criterion. The strategy will be to turn the second server on at arrival epochs, if the number of customers in the system is $\geq M^*$; and to turn the second server off at departure epochs for customers of the second server, when the number of customers in the system is $<M$ (assuming $m = M-1$). The approach to analyzing this strategy will be to write the steady-state balance equations for the system for arbitrary M. We will assume $M>0$ since $M=0$ is a normal two server problem.

Consider the simple case where no service errors and equal costs are incurred for the subsystems. The steady-state probabilities for which we want to form the balance equations will have elements

* The symbol M here represents the threshold value which is not related to the M in the standard notation of (M/G/2).

p_{nij} = Probability of n customers in the system with server 1 in state i and server 2 in state j where i=Ø or j=Ø indicates idleness and i=1 or j=1 indicates busyness for the appropriate server.

Arrival rates $_n$ as well as service rates $_{1n}$ and $_{2n}$ may be functions of system state.

Now, we can draw the state diagram (Figure I-1). The state equations can be written using the rate out equals rate in approach [White, et al, 1975],

$$\lambda_0 P_{000} = \mu_{11} P_{110} + \mu_{21} P_{101}$$

$$(\lambda_1 + \mu_{11}) P_{110} = \lambda_0 P_{000} + \mu_{12} P_{210} + \mu_{22} P_{211}$$

$$(\lambda_1 + \mu_{21}) P_{101} = \mu_{12} P_{211}$$

$$(\lambda_i + \mu_{1i}) P_{i10} = \lambda_{i-1} P_{(i-1)10} + \mu_{1(i+1)} P_{(i+1)10} + \mu_{2(i+1)} P_{(i+1)11}, \qquad 2 \leq i \leq M-2,\ M \geq 4$$

$$(\lambda_i + \mu_{1i} + \mu_{2i}) P_{i11} = \lambda_{i-1} P_{(i-1)11} + \mu_{1(i+1)} P_{(i+1)11}$$

$$(\lambda_{M-1} + \mu_{1(M-1)}) P_{(M-1)10} = \lambda_{M-2} P_{(M-2)10} + \mu_{2M} P_{M11}$$

$$(\lambda_{M-1} + \mu_{1(M-1)} + \mu_{2(M-1)}) P_{(M-1)11} = \lambda_{M-2} P_{(M-2)11} + \mu_{1M} P_{M11}$$

$$(\lambda_M + \mu_{1M} + \mu_{2M}) P_{M11} = \lambda_{M-1} P_{(M-1)10} + \lambda_{M-1} P_{(M-1)11} + (\mu_{1(M+1)} + \mu_{2(M+1)}) P_{(M+1)11}$$

$$(\lambda_i + \mu_{1i} + \mu_{2i}) P_{i11} = \lambda_{i-1} P_{(i-1)11} + (\mu_{1(i+1)} + \mu_{2(i+1)}) P_{i11}, \qquad M+1 \leq i \leq K-1$$

$$(\mu_{1K} + \mu_{2K}) P_{K11} = \lambda_{K-1} P_{(K-1)11}$$

An additional equation constraints the probabilities to sum to one,

$$P_{000} + P_{110} + P_{101} + \sum_{i=2}^{M-1} (P_{i10} + P_{i11}) + \sum_{i=M}^{K} P_{i11} = 1$$

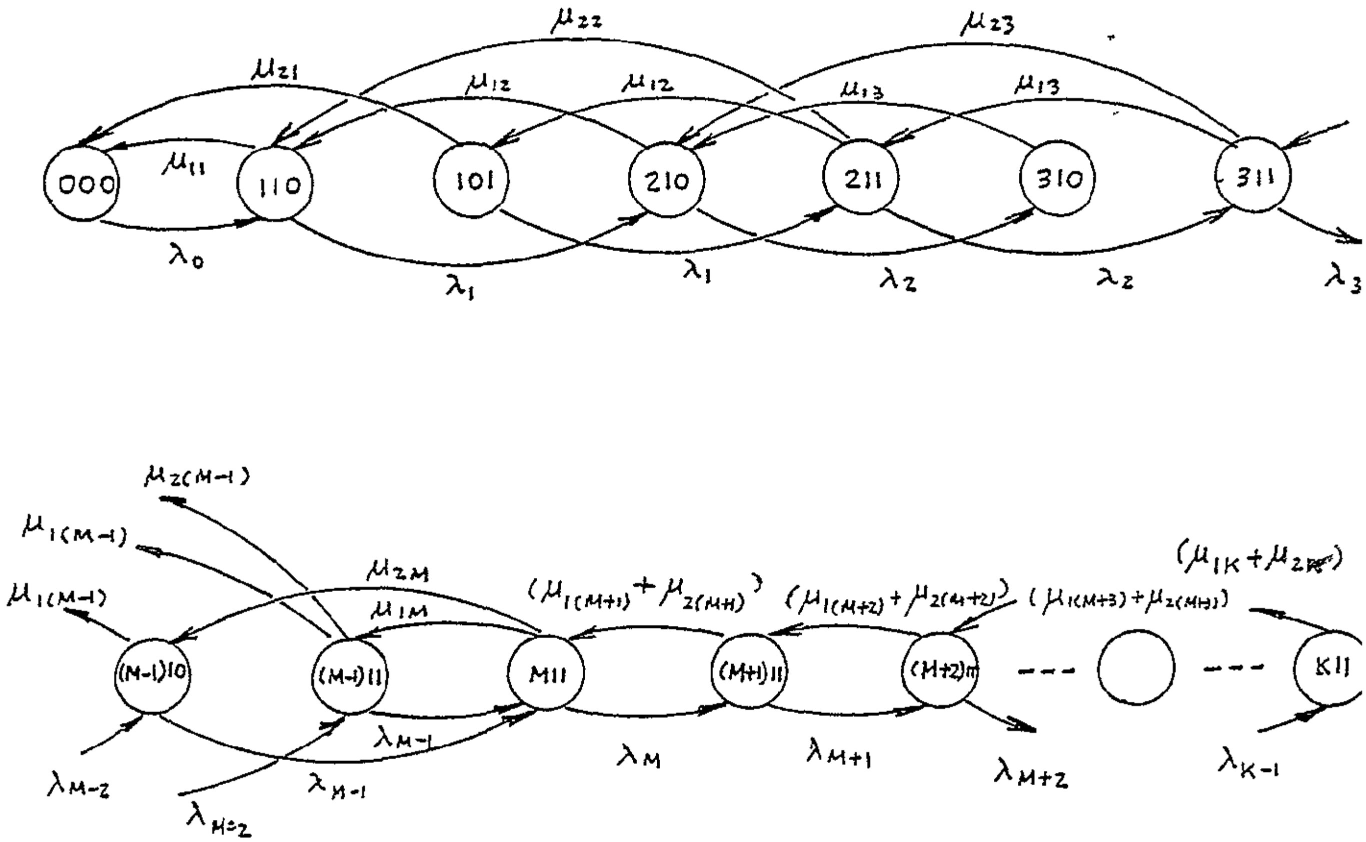

Figure I-1. State-transition-rate diagram of an (M/G/2):(GD/K/K) queue.

Solving this equation for p_{N11} and substituting into the last of the balance equations, we can arrange in matrix form to obtain

$$\begin{bmatrix} & & \\ & (K+M-1) \times (K+M-1) & \\ & & \end{bmatrix} \begin{bmatrix} P_{000} \\ P_{110} \\ P_{101} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_{(K-1)11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ (\mu_{1K}+\mu_{2K}) \end{bmatrix} .$$

and thus,

$$\begin{bmatrix} P_{000} \\ P_{110} \\ P_{101} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_{(K-1)11} \end{bmatrix} = \begin{bmatrix} & & \\ & (K+M-1) \times (K+M-1) & \\ & & \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ (\mu_{1K}+\mu_{2K}) \end{bmatrix}$$

With the state probabilities defined, we can calculate operating characteristics

$$\tilde{\lambda} = \lambda_0 P_{000} + \lambda_1 (P_{110} + P_{101}) + \sum_{i=2}^{M-1} \lambda_i (P_{i10} + P_{i11}) + \sum_{i=M}^{K} \lambda_i P_{i11}$$

$$L = (P_{110} + P_{101}) + \sum_{i=2}^{M-1} i(P_{i10} + P_{i11}) + \sum_{i=M}^{K} iP_{i11}$$

$$W = L/\tilde{\lambda} \ ;$$

$$\rho_1 = 1 - P_{000} - P_{101}$$

$$\rho_2 = 1 - P_{000} - \sum_{i=1}^{M-1}$$

Appendix II

Computer Simulation Program Structure of an (M/G/2):(GD/K/K) Queue

A Fortran simulation program has been written to simulate an (M/G/2):(GD/K/K) queue with removable second server. This is based on the Monte Carlo method of event generation and time iteration of event activity scanning. The simulation flow chart shown in Figure II-1 represents an scanning process, which starts at time = Ø, moves from one event to next, records the changes in the system at each event. The process continues until the next event is the end of the simulation, then the statistics of interest are calculated.

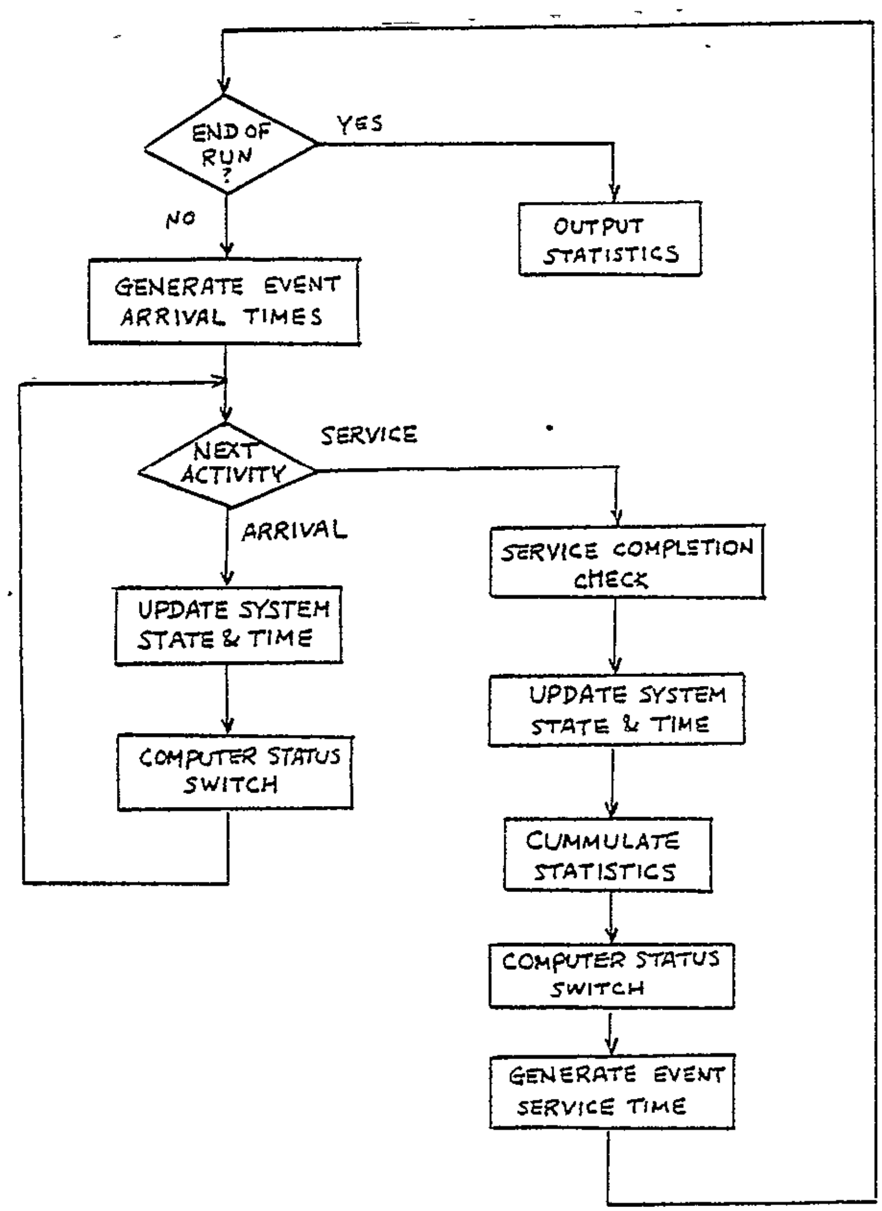

Figure II-1. Flow chart of a general (M/G/2):(GD/K/K) queue.

Appendix III

Experimental Apparatus and Software

The experimental situation is generated by a PDP-11/40 driven CRT graphics system. The computer system includes a removable-disk drive, real-time clock, and floating-point arithmetic hardware.

A spring-centered x-y joystick (Measurement Systems, Inc., Model 521), and two control levers (connected to linear potentiometers) are interfaced to the computer using a multiplexed A/D converter. The joystick is used by subjects to control aileron and elevator deflection. The control levels are used to control the speed of the simulated aircraft and to setup the appropriate control mode. The graphics system includes a display processor (with point, vector, and character generation hardware), and a vector-type display CRT (Hewlett Packard Model 1310A; 38.1 x 27.9 cm. screen size). The display processor uses direct access of the PDP-11 memory for display refresh at a rate of 30 Hz. Using this graphics system, a simulated airplane instrument panel is presented to the pilot (see Figure 3-1). A 4 x 3 numeric entry keyboard (portion of an Infoton Vistar/2 terminal keyboard) is used by the pilot for entering his responses to events occurring on the display. (See Figure III-1 for the experimental situation.) A

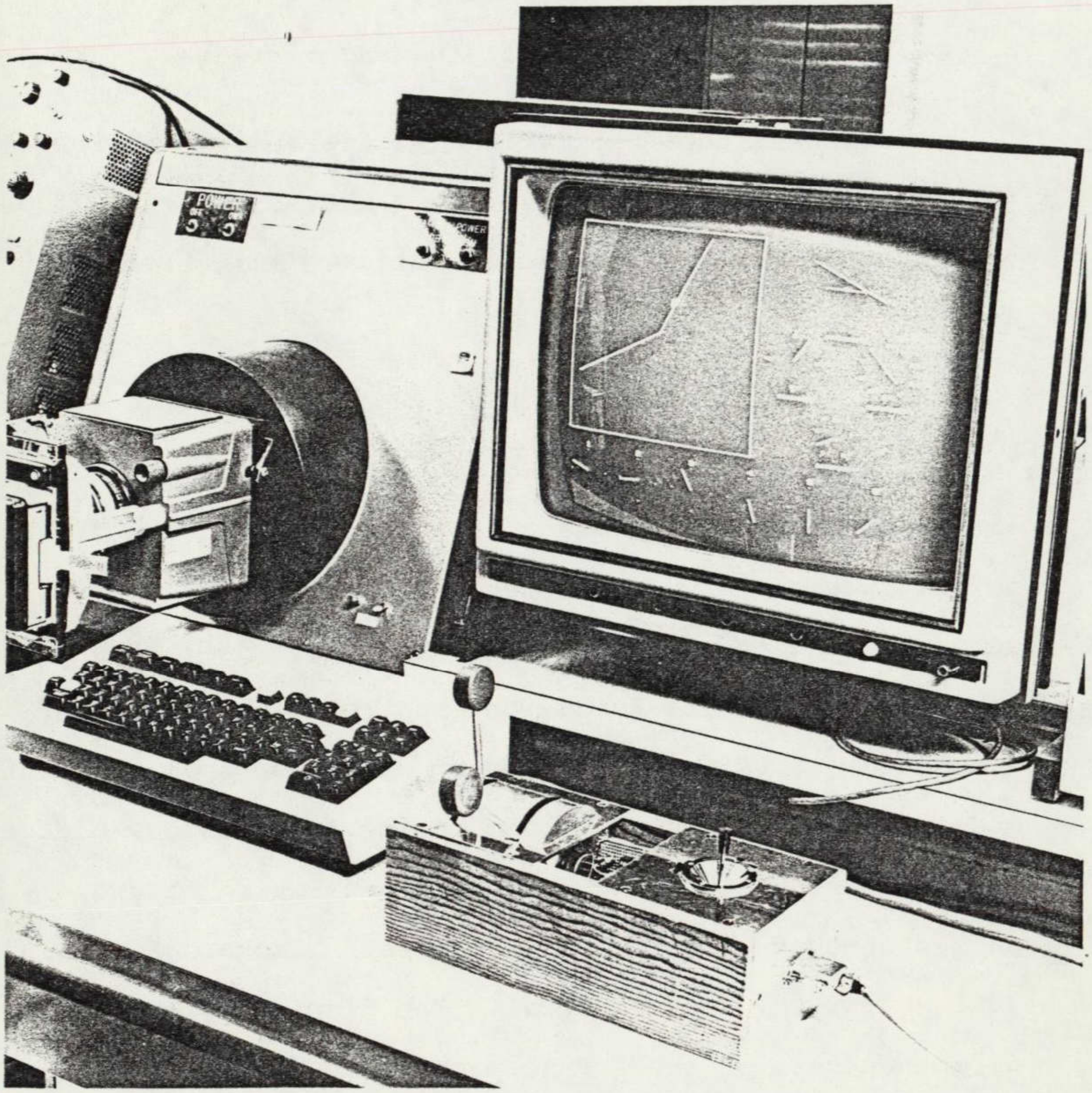

Figure III-1. Experimental situation.

for diagnostic action, one of two things can happen. If no event has occurred in the selected subsystem, a false alarm has been made, and the loop is begun again. If an event has occurred in the subsystem, the checklist diagnostic procedure is begun. If the subject makes an incorrect response in the checklist procedure, the main loop is immediately restarted. Otherwise, after the response to the event is correctly completed, the subsystem pointer is redrawn (upward), and another event is scheduled for that subsystem, before the loop is restarted.

As the main iteration loop is being executed, the real-time clock is running, and checks are made frequently to determine if it is time to perform a system states and status update. Update of the simulation state is made every Ø.5Ø second. Several things happen during a simulation update. First, a data sample is taken, and stored on the PDP-11 disk. The state of each subsystem indicator, the status of each subsystem (whether or not an event has occurred), control inputs and keyboard responses, and the states of the aircraft dynamics are sampled. Also included are autopilot status as well as computer aiding status in the aiding experiment.

Next, a schedule of events is checked, and if an event is to occur at the present time, the corresponding subsystem indicator is redrawn (downward) on the display. In autopilot mode, if an autopilot malfunction occurs, the aircraft roll angle drift is added to the aircraft dynamics (Section 3-4). Or, during autopilot restoration, distance of the aircraft from the "on-schedule" marker is measured and compared with a calculated distance to decide if the autopilot is to be re-engaged. In the event of computer aiding available, to coordinate the computer decision maker the status of subsystems has to be checked to determine if there is any system change. Upon encountering an arrival or completion of a sybsystem event, the program then calculates both the weighted sum of events and the threshold to decide if the computer should be turned on or off.

The aircraft display update is performed next. The position of the airplane is updated, as well as the position of the "on-schedule" marker, and both are redrawn on the display. If the airplane is near enough to the edge of the map currently being displayed, the map is updated to show the next portion of the course. The cockpit instruments are updated to reflect the current status of the aircraft, and redrawn. The aircraft state variables are updated, for use in the next iteration. The joystick position and the setting of the control levers are sampled. At this point, the simulation update is complete, and the main iteration loop is resumed.

When the subject has flown the airplane over the complete course (the course is completed when the airplane moves to the right of a vertical line through the target at the end of the course), or after a specified time has elapsed,whichever is desired, the simulation ends. At this time, information such as starting and finishing time of the trial, subject name, experiment identification, date, and subjective pilot comments are recorded in the data file.

Appendix IV

Aircraft Dynamics

The aircraft dynamics used in the simulation are basically those of a Boeing 707, taken from Blakelock [1965]. Several simplifying assumptions are made. The atmosphere is assumed to be at rest relative to the earth (groundspeed = airspeed). No rudder input was provided in the experimental setup. The yaw rate r for non-zero roll angles is given by $r = \frac{g}{V_T} \sin \phi$. ($V_T$ is assumed equal to u, the airspeed.) The transfer functions used are (flat-earth) approximations linearized about an equilibrium flight condition corresponding to level flight at constant altitude and speed. The short-period longitudinal transfer functions used are for pitch rate q, and z-velocity (relative to aircraft-fixed axes) component w, as a function of elevator deflection δe:

$$\frac{q}{\delta e} = - \frac{2.391s + 2.985}{0.428s^2 + 1.408s + 2.934}$$

$$\frac{w}{\delta e} = - \frac{0.031s + 2.452}{0.428s^2 + 1.408s + 2.934}$$

The lateral (roll) transfer function gives roll rate p as a function of aileron deflection δa:

$$\frac{p}{\delta a} = \frac{0.064s^3 + 0.034s^2 + 0.197s}{0.002s^4 + 0.007s^3 + 0.008s^2 + 0.019s + 0.00003}$$

(The coefficients in the above transfer functions were calculated using the formulas from Blakelock [1965], substituting the speed and altitude used--600 fps., and 40,000 ft.--where appropriate. Since the resulting dynamics were judged to be overly difficult to control, further adjustments were made to stabilize the roots of the resulting equations. The goal here was not to perfectly model the Boeing 707 dynamics, but to provide reasonable dynamics which could be learned relatively quickly.)

The angular velocities p, q, and r are projected onto earth-reference axes using the standard formulas (see Etkin [1972]), yielding $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$. These earth-reference angular velocities are integrated using numerical integration of the form $x(k+1) = x(k) + \dot{x}(k)*dt$, to give standard earth-reference angles ϕ, θ, and ψ.

The velocity components (u,v,w) relative to aircraft body-fixed axes, are approximated as follows. u is assumed equal to the airspeed. v is zero, since turns are coordinated, and yaw angle is zero. w is given by the transfer function above.

Aircraft-fixed reference velocities u and w are projected onto earth fixed axes using the formulas:

$$\dot{x}_e = u(\cos\theta\cos\psi) + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)$$
$$\dot{y}_e = u(\cos\theta\sin\psi) + w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)$$
$$\dot{z}_e = u(-\sin\theta) + w(\cos\phi\cos\theta)$$

These velocities are integrated (as the angular velocities above) to give position and altitude of the aircraft.

Appendix V

Subjective Rating Questionnaire

At the end of each experimental run, a rating questionnaire shown in Figure V-1 was given to the subject. On the rating scale following each question, subjects made a mark indicating their perception of relative effort and quality of computer aiding. These ratings were then quantified and scaled for statistical analysis. Subjects' comments were also summarized.

1) What level of effort did you have to expend?

very low	low	moderate	high	very high

2) How easy was it to interact with the computer aiding?

very difficult	difficult	reasonable	easy	very easy

3) What effect do you think computer aiding had on overall performance?

large degr.	modest degr.	slight degr.	no effect	slight impr.	modest impr.	large impr.

4) How desirable do you find computer aiding?

definitely don't like it	somewhat undesirable	doesn't matter	somewhat desirable	definitely like it

5) Other comments:

Figure V-1. Subjective rating questionnaire.

Appendix VI

Analysis of Variance

A set of analyses of variance based on the data for eight subjects was conducted. The results for each experimental measures are listed in the following tables.

Source	df	MS	F	
Control Mode (C)	2	824.131	67.806	($p < 0.005$)
Computer Aiding (A)	1	375.685	30.910	($p < 0.005$)
Arrival Rate (R)	1	365.157	30.044	($p < 0.005$)
C x A	2	99.074	8.151	($p < 0.005$)
C x R	2	6.076	0.500	
A x R	1	36.642	3.015	
C x A x R	2	0.156	0.013	

Table VI-1. Average subsystem waiting time.

Source	df	MS	F	
Adaptive Aiding (V)	1	45.244	10.496	($p < 0.005$)
Arrival Rate (R)	1	53.587	12.431	($p < 0.005$)
V x R	1	0.247		

Table VI-2. Average subsystem waiting time: adaptive effect.

Source	df	MS(x10^5)	F	
Control Mode (C)	1	70.364	4.532	($p < 0.05$)
Computer Aiding (A)	1	7.134	0.459	
Arrival Rate (R)	1	5.382	0.347	
C x A	1	1.686	0.109	
C x R	1	3.983	0.257	
A x R	1	6.891	0.444	
C x A x R	1	6.200	0.399	

Table VI-3. RMS distance error.

Source	df	MS(x10^6)	F
Adaptive Aiding (V)	1	0.008	0.004
Arrival Rate (R)	1	0.581	0.297
V x R	1	3.373	1.726

Table VI-4. RMS distance error: adaptive effect.

Source	df	MS(x10^6)	F	
Control Mode (C)	1	5.215	4.423	($p < 0.05$)
Subsystem Presence (R)	1	8.506	7.215	($p < 0.025$)
C x R	1	0.052	0.044	

Table VI-5. RMS distance error: effect of subsystem presence.

Source	df	MS	F	
Control Mode (C)	1	0.050	20.103	($p < 0.005$)
Computer Aiding (A)	1	0.001	0.522	
Arrival Rate (R)	1	0.008	3.067	
C x A	1	0.002	0.834	
C x R	1	0.003	1.276	
A x R	1	0.000	0.021	
C x A x R	1	0.000	0.004	

Table VI-6. RMS roll angle.

Source	df	MS	F	
Control Mode (C)	1	0.011	7.204	($p < 0.025$)
Presence of Subsystem (R)	1	0.000	0.008	
C x R	1	0.002	1.605	

Table VI-7. RMS roll angle: effect of subsystem presence.

Source	df	MS	F
Adaptive Aiding (V)	1	0.009	2.557
Arrival Rate (R)	1	0.001	0.205
V x R	1	0.007	1.899

Table VI-8. RMS roll angle: adaptive effect.

Source	df	MS	F	
Control Mode (C)	2	0.429	28.085	($p < 0.005$)
Computer Aiding (A)	1	1.244	81.381	($p < 0.005$)
Arrival Rate (R)	1	0.098	6.425	($p < 0.025$)
C x A	2	0.014	0.925	
C x R	2	0.009	0.568	
A x R	1	0.001	0.053	
C x A x R	2	0.006	0.406	

Table VI-9. Empirical server occupancy.

Source	df	MS	F	
Arrival Rate (R)	1	0.459	102.115	($p < 0.005$)
Adaptive Aiding (V)	1	0.009	1.962	
R x V	1	0.000	0.018	

Table VI-10. Empirical server occupancy: adaptive effect.

Source	df	MS	F
Adaptive Aiding (V)	1	141.414	0.489
Arrival Rate (R)	1	586.959	2.031
V x R	1	19.861	0.069

Table VI-11. Average autopilot malfunction waiting time.

Source	df	MS	F	
Control Mode (C)	2	0.940	35.963	($p < 0.005$)
Computer Aiding (A)	1	2.074	79.323	($p < 0.005$)
Arrival Rate (R)	1	0.106	4.054	($p < 0.05$)
C x A	2	0.119	4.564	($p < 0.025$)
C x R	2	0.002	0.071	
A x R	1	0.021	0.792	
C x A x R	2	0.010	0.394	

Table VI-12. Subjective effort ratings.

Source	df	MS	F	
Adaptive Aiding (V)	1	0.009	0.236	
Arrival Rate (R)	1	0.549	13.723	($p < 0.005$)
V x R	1	0.002	0.057	

Table VI-13. Subjective effort ratings: adaptive effect.

Source	df	MS	F	
Control Mode (C)	2	1.018	27.114	($p < 0.005$)
Arrival Rate (R)	1	2.258	60.109	($p < 0.005$)
C x R	2	0.042	1.130	

Table VI-14. Subjective ratings of the effectiveness of computer aiding.

Source	df	MS	F
Adaptive Aiding (V)	1	0.045	0.416
Arrival Rate (R)	1	0.242	2.230
V x R	1	0.020	0.185

Table VI-15. Subjective ratings of the effectiveness of adaptive aiding.

Source	df	MS	F	
Control Mode (C)	2	0.445	6.601	($p < 0.005$
Arrival Rate (R)	1	2.168	32.123	($p < 0.005$)
C x R	2	0.160	2.375	

Table VI-16. Subjective ratings of the desirability of computer aiding.

Source	df	MS	F	
Adaptive Aiding (V)	1	0.385	4.499	($p < 0.05$)
Arrival Rate (R)	1	1.492	17.436	($p < 0.005$)
V x R	1	0.009	0.103	

Table VI-17. Subjective ratings of the desirability of adaptive aiding.

Source	df	MS	F	
Control Mode (C)	2	0.324	3.666	($p < 0.05$)
Arrival Rate (R)	1	0.163	1.846	
C x R	2	1.276	14.416	($p \leq 0.005$)

Table VI-18. Subjective ratings of the ease to interact with computer aiding.

Source	df	MS	F
Adaptive Aiding (V)	1	0.130	1.579
Arrival Rate (R)	1	0.088	1.071
V x R	1	0.000	0.002

Table VI-19. Subjective ratings of the ease to interact with adaptive aiding.

Appendix VII

Simulation Flow Diagram for a Flight Management Situation

The flow chart of Figure VII-1 simulates a flight management situation represented as an (M/G/2):(PRP/K/K) queue with removable second server. The preemptive resume priority discipline of control service is used. The presence and absence of control tasks is also implemented to represent the various control modes (such as manual or autopilot malfunction modes, etc.) in the flight management situation.

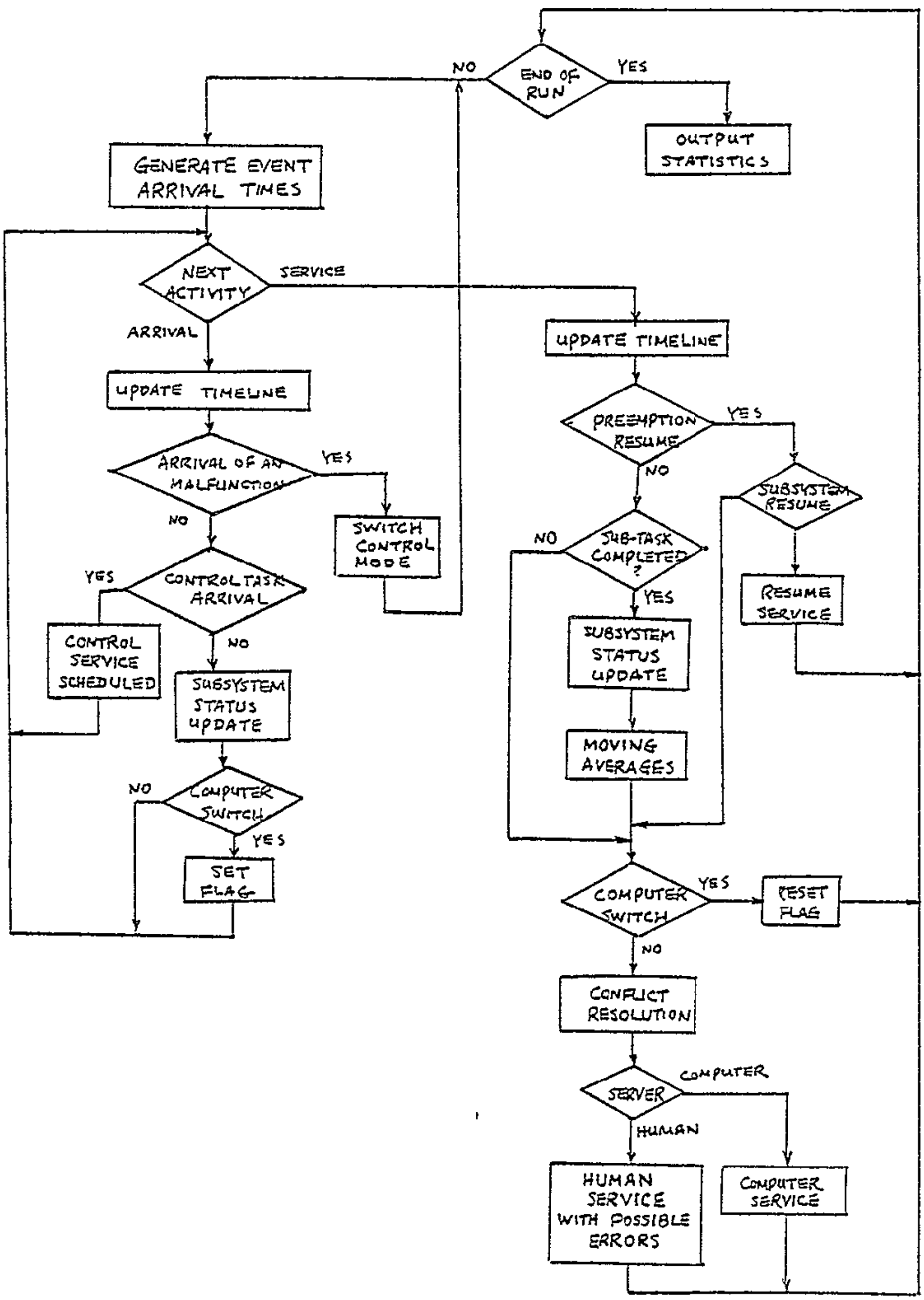

Figure VII-1. Flight management simulation flow diagram.

VITA

Yee-yeen Chu was born in [redacted] on [redacted]. He attended the National Taiwan University where he earned the B.S. degree in Mechanical Engineering in June 1971. He was commisioned a second lieutenant in the Chinese Army during the period of 1971 to 1973. In August 1973 he came to United States and earned the M.S. degree in Mechanical Engineering from Clemson University in December 1974. He then went to the University of Illinois and earned the Ph.D. degree in Mechanical Engineering in August 1978.