D45

N79-15633

PILOT DECISION MAKING IN A COMPUTER-AIDED FLIGHT MANAGEMENT SITUATION*

Yee-yeen Chu and William B. Rouse

Department of Mechanical and Industrial Engineering
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

SUMMARY

An experimental representation of a computer-aided multi-task flight management situation has been developed. A computer aiding program was implemented to serve as a back-up decision maker. An experiment was conducted with a balanced design of several subject runs for different workload levels. This was achieved using three levels of subsystem event arrival rates, three levels of control task involvement, and three levels of availability of computer aiding. Experimental results compared quite favorably with those from a computer simulation which employed a $(M/E_k/2):(PRP/K/K)$ queueing model. It was shown that the aiding had enhanced system performance as well as subjective ratings, and that the adaptive aiding policy further reduced subsystem delay.

INTRODUCTION

As aircraft become more complicated and greater demands and better performance are being required of 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.

Equiped with autopilot and airborne computers performing automatic navigation, guidance, energy calculations, flight planning, information management, etc., the next-generation of aircraft are quite likely to be capable of carrying out all phase of flight automatically. However, the human pilot is likely to reamin 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.

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

676
PAGE INTENTIONALLY BLANK

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. [1]. 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 when 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. The problem that arises is how to allocate responsibility between the pilot and the computer for a subset of all tasks.

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 [2]. 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 was developed and will be discussed in the next section.

APPROACH

The pilot in the automated flight management system described earlier has a variety of tasks to perform. As the number and variety of tasks increases, the workload of the pilot is increased. It is essential to appropriately allocate his attention and effort among the tasks. He may be in a situation that he wants both to monitor the tasks often enough to reduce growing uncertainty and risk, and to perform a task quickly and accurately to lessen the cost involved in the delay of action. This issue is being investigated by Greenstein and Rouse [3]. To simplify the issue, the pilot 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 unequivocally present themselves. The pilot 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 adapt the same strategy either by being hard-wired or learning from the pilot. Now we may look at the multi-task decision making as a queueing system 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).

In the queueing model, each server is characterized by his observation of system state, his perceptions of event occurrences, of event arrival rates and of event service rates. Combining the above information and the system cost criteion allows the model to predict system performance measures such as event delay statistics and server occupancy which is fraction of time the server is busy.

A convenient cost criterion, in terms of a stationary expected cost structure, includes waiting cost, service cost, and switching cost. When the computer service cost and switching cost may be negligible, the optimal policy is to have the computer on all the time. However, it is more likely that the human will be better at performing the task but not have sufficient time to do all the tasks. Also evidence of vigilance and warm-up decrements suggests that there is an acceptable workload range that sustains performance on long tasks. Thus we may want to seek a policy for computer aiding such that a minimum waiting cost is achieved while maintaining a specified workload level.

Based on results from literature [4], we will advocate the use of the stationary expected cost policy, subject to minimizing deviation from acceptable pilot workload, for computer on-off of the following form: turn the computer on at arrival epochs when $N = c_1 n_1 + c_2 n_2 + \dots + c_K n_K > M$, and turn it off when $N < m$, where $c_1, c_2, \dots, c_K$ are cost rates assessed according to relative priorities and n_k is the number of events waiting in the subsystem k. This policy (i.e., M and m) should vary as the system variables vary. Specific values of M and m have to be determined for various levels of traffic demand (i.e., event arrival rates), server performance and task complexity (i.e., service rates and probabilities of errors). An appropriate approach to implement the adaptive policy is to set up a table of stationary control policies beforehand and to employ a table look-up along with on-borad estimation of system variables.

To obtain the optimal stationary policy, i.e., to determine the values of M and m, a computer simulation was performed. Poison arrivals and Erlang service time distributions for subsystem 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. 2) Two levels of arrival rates were assumed, i.e., low arrival (at 0.0167 events per second) and high arrival (at 0.0333 events per second). 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 server occupancy for pilot of = 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. If workload is the primary consideration, these are threshold values which the computer should employ to adapt to both the subsystem arrival rate and the control task involvement.

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

THE EXPERIMENT

Two experiments are to be discussed here. A brief review is given of an experiment previously reported by Walden and Rouse [5] 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 [6] used a PDP-11 driven CRT graphic system to represent a cockpit-like display to an experimental subject. The display shown in Figure 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 shcedule.

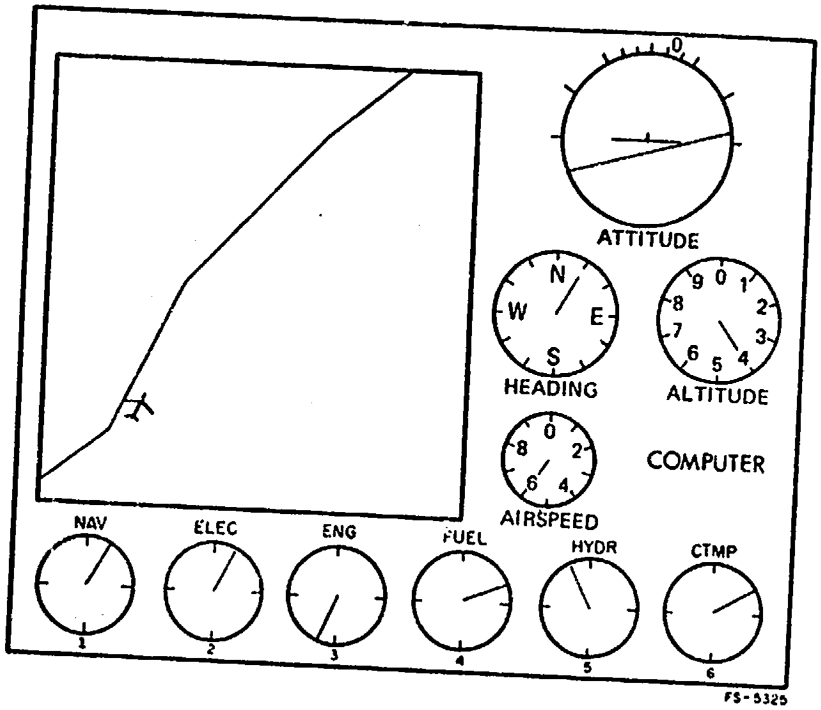

Figure 1. The Flight Management Situation.

In the manual control mode, the pilot controlled the pitch and roll of Boeing 707 aircraft dynamics with a joystick. Another control stick controlled 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 1) to which he wished to respond, the subject selected that subsystem via a 4x3 kevboard. The display shown in Figure 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 seleted the branch with his keyboard. After completing the last level of the tree, the action was completed and the display shown in Figure 1 returned, with the subsystem information or diagnostic check complete.

Using the experimental situation, an experiment was performed by Walden [5] 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.

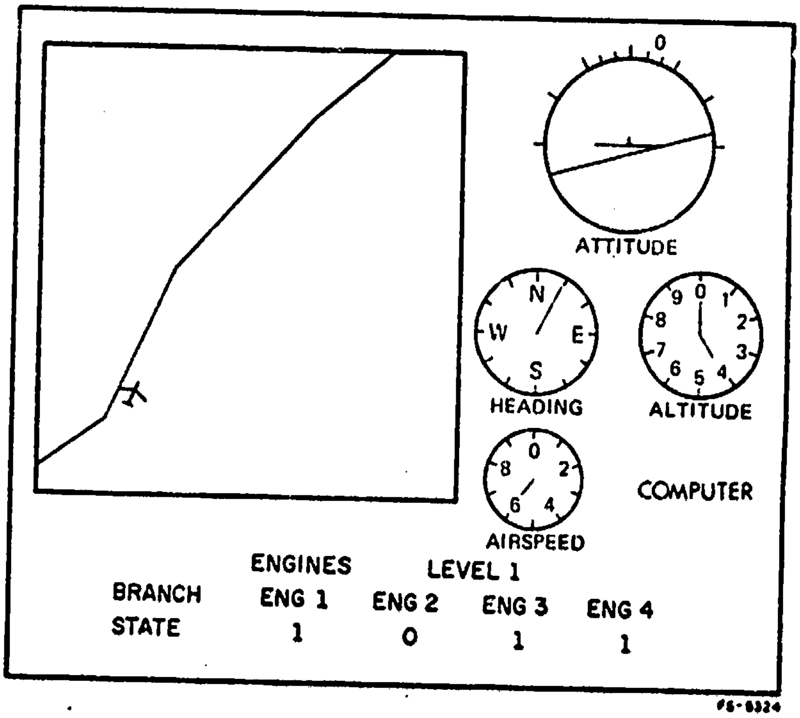

Figure 2. Display When Pilot Had Reacted To an Event in Engine Subsystem.

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 probably 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, 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 a hierarchical check-list procedure. In the experimental situation shown in Figure 3, The 'NAV' symbol over the navigation dial flashed, if the computer decided that an event had occurred and was waiting to be serviced in the navigation system. This was to tell 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 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 status. For the purpose of the experiment reported here, however, the communication channel from the pilot to sybsystems was predefined. 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 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.

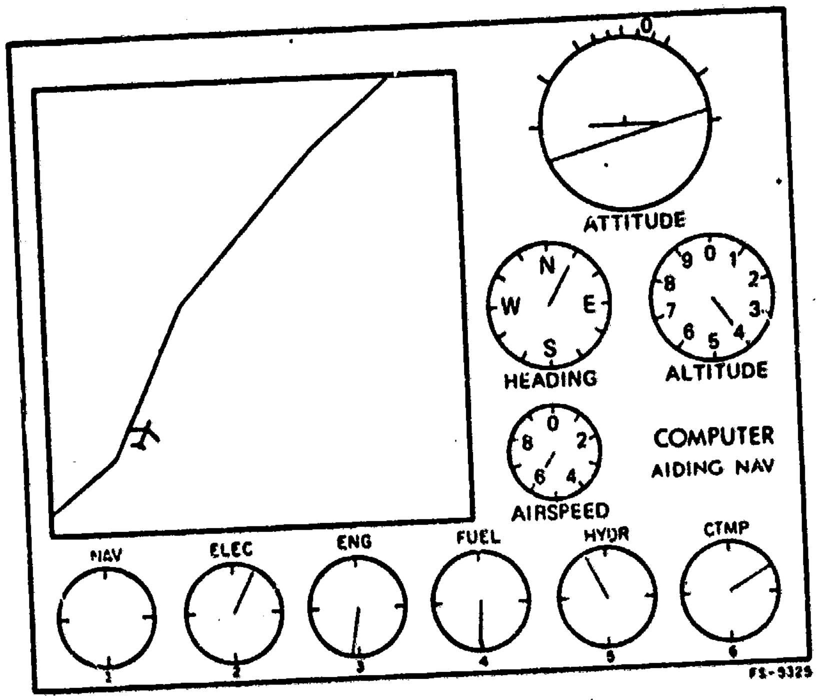

Figure 3. Display When the Computer Is Servicing Navigation System.

While the computer had to constantly check the pilot's action to avoid a conflict, 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.

An experiment based on the experimental representation described above was conducted. Eight trained subjects, all of them male students in engineering, participated in a balanced sequence of sixteen experimental runs (see Table 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 was filled out by the subject.

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. Design of Experiment.

	Subject 1	Subject 2	Subject 3	Subject 4
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

For the experiment 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 posible, 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 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 the 800-feet oval of the on-schedule circle, and the autopilot mode was restored. The autopilot malfunction happened relatively infrequently, based on a Poison distribution with mean inter-arrival time of 160 seconds.

After the pilot detected the autopilot malfunction, he would have to devote a major portion of his attention to the control task, leaving subsystem tasks less attended, while 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 malfunction), 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) 3-D RMS and average flight course 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 is incurred. The results based on the analyses of variance are discussed in the next section.

RESULTS

The subsystem event waiting times averaged across subjects for the various task situations are shown in Figure 4. An analysis of variance conducted showed that among the statistically significant factors (at the .05 level) are the three experiment variables, i.e., the control mode, the subsystem arrival rates, and the computer aiding. As expected, the subsystem waiting time increased as the subsystem arrival rate increased, as the control involvement increased, and when no computer aiding was provided. A

ORIGINAL PAGE IS
OF POOR QUALITY

separate test showed that the adaptive policy was also significant, i.e., the adaptive aiding further reduced the subsystem waiting time beyond the fixed-threshold aiding, even though the adaptive policy was only effective during a small portion of time in the experiment.

The subjective ratings of the level of effort across subjects are shown in Figure 5 Factors of significance include all three experiment variables. As expected, the perceived level of effort increased as control involvement increased, as subsystem arrival increased, and as computer aiding was removed. However, a separate test showed that the effect of the adaptive policy was not significant, probably because the adaptive policy was employed rather infrequently, and 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 then usual.

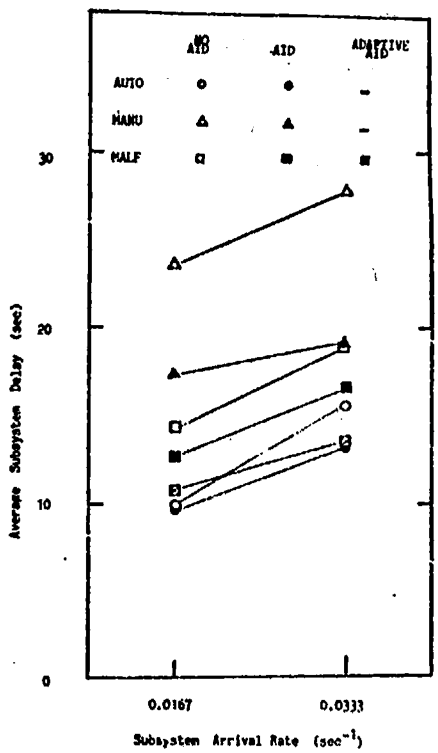

Figure 4 Average Subsystem Delay.

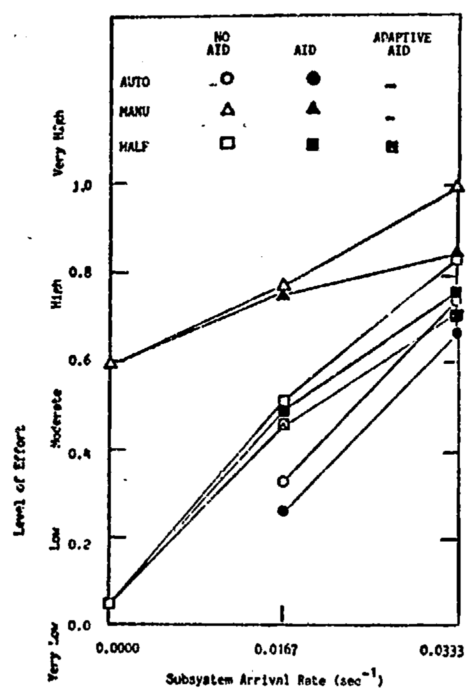

Figure 5 Subjective Ratings of Effort.

The RMS course error across subjects is shown in Figure 6. The analysis of variance showed that only control mode had an effect on the control error. No consistent variation in the course error was shown as subsystem arrival rate or aiding situation varied. The lower course RMS error for the autopilot malfunction mode probably resulted from subject's more intense attention to the control task in the case of malfunction.

The RMS roll angle across subjects is shown in Figure 7. Also, only control mode had a significant effect on the control input. The subjects were found to use more extreme control actions and more attention to fulfill the malfunction task requirements. Sumarizing the above, 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 requirement, such as computer aiding or highly reliable subsystems, seem to improve only subsystem performance. The possible reason for this is that the control task perempts subsystem tasks, and thus, the control task inefficiency is likely to affect the performance of subsystem tasks; the reverse is not true.

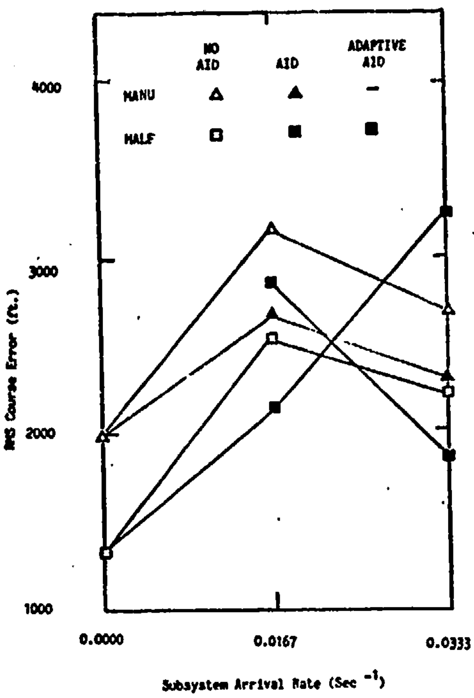

Figure 6. RMS Course Error.

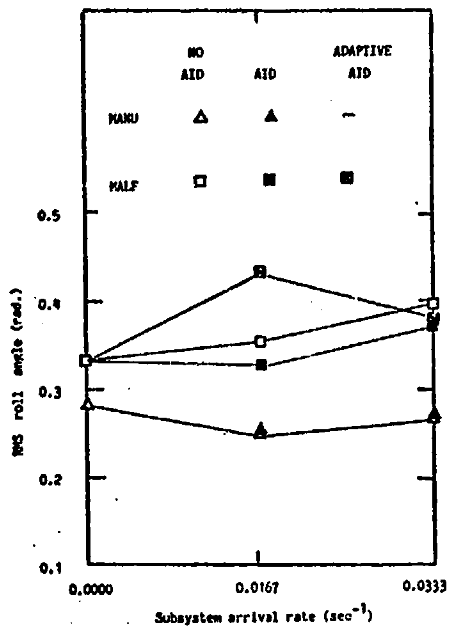

Figure 7. RMS Roll Angle.

Subjective ratings of three aspects of computer aiding were also determined: effectiveness, desirability of the aiding, and ease of interaction with the aiding. The results indicate that the aiding was considered easy to interact with and desirable by the subjects. Its effect on performance improvement was perceived to be from moderate to large. The subjects perceive the aiding to be relatively more effective and more desirable with a high subsystem arrival rate or a high control involvement situation. They, however, did not feel that it was more difficult to interact with the aiding in those situations. In fact, all the subjects were quite in favor of both the aiding scheme used in the experimental situation and the general computer aiding idea. More analyses of performance measures are discussed by Chu in his thesis [7].

The empirical data were compared with simulation results from the queueing model of pilot decision making in computer aided situation discussed earlier. This allowed an evaluation of the model's ability to represent the

ORIGINAL PAGE IS
OF POOR QUALITY

given situation. The comparison of subsystem waiting statistics is shown in Table 2.

Table 2. Comparison of Waiting Time.

Arrival Rate	Aiding Type	Mean Model	Mean Data	Standard Deviation Model	Standard Deviation Data
Autopilot Mode					
Low	No Aiding	9.73	9.71	5.39	6.04
Low	Aiding	9.34	9.82	4.30	5.13
High	No Aiding	14.71	15.79	13.46	14.21
High	Aiding	13.79	13.16	12.00	7.43
Manual Mode					
Low	No Aiding	20.13	23.62	16.24	23.53
Low	Aiding	17.56	17.17	10.26	11.31
High	No Aiding	32.87	27.81	45.51	28.64
High	Aiding	19.58	19.19	11.85	12.17
Autopilot Malfunction Mode					
Low	No Aiding	12.00	14.25	8.85	13.81
Low	Aiding	11.13	12.84	6.79	10.52
Low	Adaptive Aiding	10.25	10.68	4.91	5.52
High	No Aiding	17.47	19.03	18.96	21.16
High	Aiding	13.66	15.52	8.52	11.55
High	Adaptive Aiding	12.32	13.25	7.10	8.33

In the model, a Poison distribution of control event arrivals and an Erlang distribution of control service times with shape parameter k=2 were assumed. To generate the results in Table 2, the values of 0.1 sec^{-1} (in manual mode) and 0.16 (in malfunction mode) were used as mean control arrival rates, and 0.47 and 0.34 as mean control service rates. These values were obtained by analyzing subject's aileron control input and, serve as a first approximation.

The results compare reasonably well. All parameters in the model were empirically measured and no adjustments were made. The model predicts performance in autopilot mode very well. A better estimate of control task parameters will surely improve the model accuracy in manual control and autopilot malfunction modes.

CONCLUSION

The experimental results show that all the experimental variables, i.e., the subsystem arrival rates, the control task involvement, and the availability of computer aiding, were statistically significant in terms of affecting the performance measures of interest, mainly, the subsystem delays, and subjective effort ratings. It was shown that the aiding enhanced system performance in terms of subsystem average delays and subjective effort ratings. The adaptive aiding policy was shown to further reduce subsystem waiting time.

The queueing model fits the experiment result reasonably well. Further exploration of control task preemption is needed to improve model accuracy. The model also provides the capability to predict the server occupancy for different task situations. Included in the future work will be a test of the correlation between this server occupancy measure and the subjective effort ratings to determine if this measure may effectively serve as a workload indicator.

Finally, the computer-aided flight management situation will next be implemented in an aircraft simulator where regular pilots will be used as subjects.

REFERENCES

1. Wempe, T.E., "Flight Management - Pilot Procedures and System Interfaces for the 1980 - 1990's", AIAA Paper No. 74-1297, AIAA Life Science and System Conference, November 1974.

2. Chu, Y. and Rouse, W.B., "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.

3. Greenstein, J.S. and Rouse, W.B., "A Model of Event Detection in Multiple Process Monitoring Situations", Proceedings of the Fourteenth Annual Conference on Manual Control, April 1978.

4. Bell, C.E., "Optimal Operation of an M/G/1 Priority Queue with Removable Server", Operations Research, 21-6, 1973, pp. 1281-1290.

5. Walden, R. S. and Rouse, W.B., "A Queueing Model of Pilot Decision Making in a Multi-task Flight Management Situation", Proceedings of the Thirteenth Annual Conference on Manual Control, pp. 222-236, June 1977.

6. Rouse, W.B., Chu, Y., and Walden, R.S., "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.

7. Chu, Y., Adaptive Allocation of Decision Making Responsibility between Human and Computer in the Multi-task Situations", Ph.D. Thesis in progress, University of Illinois, Urbana, Illinois.