D22

N79-17497

A QUEUEING MODEL OF PILOT DECISION MAKING IN A MULTI-TASK FLIGHT MANAGEMENT SITUATION*

R. S. Walden and W. B. Rouse

ABSTRACT

Allocation of decision making responsibility between pilot and computer is considered and a flight management task, designed for the study of pilot-computer interaction, is discussed. A queueing theory model of pilot decision making in this multi-task, control and monitoring situation is presented. An experimental investigation of pilot decision making and the resulting model parameters are discussed.

INTRODUCTION

Automation has found increasing use in aircraft, not only with respect to mechanical systems, but with respect to jobs normally thought of as belonging to the airplane pilot. As airplane numbers increase and navigation becomes more complex, the trend will certainly continue. Wempe [1] suggests that an advanced automated navigation system will become necessary in the future, because the required control performance will exceed human capability. However, he presents two reasons why completely automatic control of aircraft is unlikely in the near future: a machine required to replace a pilot would be quite costly; furthermore, machines do not adapt well to unanticipated events. Control of aircraft thus will be the task of a man-machine system.

With the recent rapid expansion of digital technology (especially microprocessors), the words "man-machine" and "man-computer" have become practically synonymous. The computer can have several roles in an airborne man-computer system. Special-purpose processors can be dedicated to such things as autopilot and navigation aids, displays, communication systems, or subsystem monitoring; in such applications, the computers would serve merely as digital controllers of the various devices. The pilot's workload would thus be reduced as the computers handle much of the low-level information processing.

Or, to go a step further, the computer could have a more general role. The computer might have inputs from a variety of aircraft devices and subsystems, and increased control capabilities as well. The computer would then be expected to initiate actions of its own (i.e., make decisions), based on the inputs it receives. The pilot's workload would be reduced at a higher level, by the computer sharing in decision-making tasks. It is this latter concept of the role of the computer in airborne systems which is the concern of the research described herein.

A computer with significant decision-making responsibilities in the flight-management task can create potential problems, however. One of the most crucial problems is that of allocation of decision-making responsibility between pilot and computer [2]. Allocation can either be static (situation-independent), or dynamic (situation-dependent).

Consider first a system in which allocation of responsibilities is static. There is a pre-defined, non-overlapping set of responsibilities for each decision-maker (man and computer). Such a system has one desirable feature in that no confusion is possible over who should perform a given task. However, there are several serious drawbacks to the static allocation of responsibility. These include:

1. Poor utilization of system resources due to each decision maker being prohibited from performing the other decision maker's tasks, even when one decision maker is overloaded and one is idle.

2. Difficulty for the pilot when unforeseen circumstances force him to take over tasks for which he normally has no responsibility.

3. Possible pilot resentment at not being allowed to perform certain tasks, especially when he has nothing else to do.

For the above reasons, a dynamic allocation of responsibility might be more appropriate. Simulation studies of this alternative by Rouse [3] support this conclusion. Succinctly, the idea is that the server who is most able, in a given situation, will assume responsibility for a task.

Dynamic allocation of responsibility is not without its drawbacks, however. The most important problem to be solved is that of communication between pilot and computer. This is necessary to avoid conflicts which occur when the two decision makers act independently, resulting in degraded performance. (Rouse [3] presents simulation results that illustrate this effect). A possible solution to this problem is to use the computer as a backup for the pilot, to be turned on when the pilot's workload becomes too great and/or his performance degrades.

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

The man-computer system is thus seen as one in which both the pilot and computer monitor each other--the pilot supervises and monitors the computer's activities, for the purpose of insuring that the computer is performing adequately, and that no malfunction has occurred; the computer monitors the pilot to determine when he is overloaded and needs assistance. Such a system would hopefully minimize conflict and avoid the difficulties discussed above.

In order for the computer to be able to tell when the pilot needs assistance, it must first have a method for determining the pilot's workload, in terms of task demands. Then, it needs a criterion for deciding if it should request responsibility for any tasks. Chu and Rouse [4] have proposed a formulation of this problem. To apply this formulation to any specific situation, the first step is development of a model of the unaided performance of the human with respect to tasks typical of those encountered in the situation of interest.

The purpose of this report is to present the results of pursuing this first step in the context of flight management. An experimental situation which was developed to study pilot performance in simulated flight conditions is described. Alternative models that describe pilot decision making in this flight management situation are presented. Finally, an experiment is discussed whose purpose was investigation of the suitability of the alternative models.

THE TASK

The pilot is presented with a simulated airplane instrument panel, drawn on a CRT (see Figure 1). The display includes standard aircraft instruments, such as artificial horizon, altimeter, heading and airspeed indicators. Also displayed is a map which indicates the course the airplane is to follow. An airplane-shaped symbol, superimposed on the map, indicates the airplane's actual position. A small circle moves along the mapped course at a constant speed of 600 fps (183 m/s) and indicates the position the aircraft should have for it to be on course and schedule. Near the lower edge of the display, several dials are shown, which represent (abstractly) gauges for such things as fuel, electrical, or hydraulic subsystems. The gauge pointers move with apparently random motion, generated by passing Gaussian white noise through a second-order sampled-data filter.

The pilot controls Boeing 707 aircraft dynamics as discussed by Blakelock [5]. He controls the aircraft pitch and roll dynamics with a joystick. Another control stick controls the airspeed. The pilot's task is to fly the airplane along the mapped route, maintaining a fixed altitude and stable pitch and roll attitude. The pilot is supposed to keep as close to the "on-schedule" dot, and as close to the mapped course, as possible, while holding the aircraft at 40,000 ft. (12,192 m.) altitude.

In addition to flying the airplane, the pilot monitors the subsystem indicators for possible events. (An event has occurred when the pointer motion slows and stops, pointing downward as shown for subsystem 4 in Figure 1.) When the pilot thinks an event has occurred, he enters the corresponding number on the keyboard. (If more than one event has occurred, the pilot chooses the higher-priority, or lower subsystem number, event.) The display shown in Figure 2 then appears.

On this new display, the subsystem dials have been replaced with two rows of numbers, one labelled 'BRANCH', the other labelled 'STATE'. This corresponds to the first level of a checklist associated with the subsystem which was selected. The pilot looks for a branch with a state of '0' and enters the branch number on the keyboard. If the checklist for that branch continues, the next checklist level is displayed. When the end of the checklist is reached, the subsystem indicators reappear, with the subsystem indicator motion restored. These actions simulate the checklist procedures a pilot might go through when performing tasks in a real aircraft.

The apparatus used for the experiments, the simulation software used to produce the above task, a description of the aircraft dynamics, and an explanation of the subsystems portion of the simulation are discussed in more detail in Walden's thesis [6].

THE MODEL

The role of the human operator in complex man-machine systems is changing from that of in-loop controller to that of supervisor or monitor of automatic, possibly computer operated, controls. In such a role, it is the human's monitoring performance which is of major interest. The assumption is that essential control tasks will be performed adequately, perhaps at the expense of monitoring performance.

Thus, the goal of the modeling effort reported here is to predict the subsystem task performance, measured in terms of mean and variance of subsystem event waiting time, as a function of task parameters 1) event arrival rate, and 2) control task difficulty.

Existing models prove inadequate in describing the multi-task flight management situation discussed here. Control models, for example the Kleinman, Baron, and Levison optimal control model [7] (and its many extensions), are concerned only with predicting control performance; a model useful in predicting subsystem service performance is sought. Instrument-monitoring models do not adequately describe the multi-task flight management situation, either. The instrument-monitoring model of Senders [8] focuses on details of instrument scanning, such as instrument fixation frequencies and sequences. That of Carbonell [9] is concerned with sampling policy based on instrument priorities. Neither type of model is concerned with subsequent actions resulting from instrument observations.

A queueing theory model of the multi-task situation is proposed. The monitoring task is modeled as a single-server queueing system, with subsystem events as customers. The monitoring task is, in fact, a queueing system, and it seems natural to use a queueing theory description for it. The proposed model is a simulation model; an appropriate analytical solution is not available. (See Chu [10] for explanation of the basic operation of the queueing simulation.)

In the first modeling attempt (hereafter referred to as Model 1), the monitoring task is described as a $M/E_k/1$:NPRP/N/N queueing system. (See White, Schmidt, and Bennett [11] for definitions of the standard queueing terminology employed here.) The control task is not explicitly modeled. Instead, the control task is included implicitly in the service time distribution parameters (i.e., for the Erlang distribution, the mean and shape parameter k), which change with increased map complexity (due to the control task). Computer simulation of this queueing system provided preliminary data useful in making later improvements, but did not match the experimental data well under low traffic conditions. (See RESULTS)

The second refinement of the model (referred to as Model 2) attempts to account for the impact of the control task on the subsystem task performance. This is done by using a higher effective arrival rate for subsystem events:

$$\lambda_{effective} = \lambda_{actual} + \Delta\lambda$$

where $\Delta\lambda$ is the increase in arrival rate needed to match empirical subsystem event waiting time. The control task is thus incorporated into the monitoring portion of the task as increased event arrivals in the subsystem queues. Using the same simulation program as before, subsystem event arrival rate was varied until the mean waiting times predicted by the simulation matched the experimentally observed waiting times. This model provides a good prediction of the waiting time distributions for individual subsystems (see RESULTS). However, in terms of the physical system, the $\Delta\lambda$ parameter is admittedly unappealing

With the third refinement, the model (Model 3) approaches the real situation most closely. The control task is incorporated as a special queue; customers arriving in the control task queue can preempt the service of a subsystem event. Service of the subsystem event will then be resumed at the point of interruption after the control action is complete. This queueing system is described as a $M/E_k/1$:PRP/N/N queue.

One of the problems with this third model is defining a "customer" for the control task queue. In fact, it is displayed error (more specifically, increments of displayed error) which queues (accumulates) for attention; however, the size of the incremental "customer" is unknown. A reasonable approach is to consider the assumption made earlier that the control task customer preempts subsystem service; i.e., whenever a control task customer is present, subsystem monitoring (or event service) is immediately preempted, and the control task served. A customer in the control task queue could then be defined (loosely) as a "significant" or "action-evoking" amount of display error; when a significant error is present, subsystem service will be preempted and control inserted to null the error. A simple way to indirectly measure the frequency with which "significant" errors arrive (queue) is to measure the frequency with which displayed errors are serviced, i.e., the frequency at which control actions are inserted to null them.

To estimate the frequency of control responses, the number of separate control actions performed by the pilot can be counted, and from this number, an arrival rate for control task customers can be obtained. To obtain an estimate of the average service rate for a customer in the control task queue, the control task service rate parameter in the simulation program can be varied until a good fit to experimental data is obtained.

To account for the increased waiting time incurred from errors such as false alarms and incorrect actions, modifications to the basic simulation model were made. False alarms are accounted for by using another queue, with event (false alarms) arrival rate equal to the arrival rate of false alarms in the experiment. For the task discussed here, the service time used for a false alarm is one-fifth the mean event service time, since only the first of five levels of the checklist is shown when a false alarm occurs.

Incorrect actions are accounted for in the simulation by first estimating the probability of an incorrect action while servicing a subsystem event. This is done simply by dividing the number of incorrect actions by the number of events. Based on this rate, incorrect actions are simulated. For the monitoring task discussed here, the penalty for an incorrect action is an increase in waiting time equal to one-half the mean service time for subsystem events. (On the average, if incorrect actions occur randomly, the pilot will be halfway through a checklist procedure, and thus waste one-half of the average service time.) The waiting-time penalty could be even greater, if after an incorrect action (and the subsystem indicators are returned to the display), the pilot detects and services an event in a higher-priority queue. This is also accounted for in the simulation model.

AN EXPERIMENT

Using the experimental situation described earlier, an experiment was performed to study subject "pilot" performance in a multi-task flight management situation. The two independent variables in the experiment were the inter-arrival times of subsystem events, and the difficulty of the flight path to be followed by the pilot.

In the first part of the experiment, only the subsystem monitoring task was considered. An "autopilot" kept the airplane on course, coincident with the "on-course indicator" marker. 10-minute trials, using 30, 60, and 90 second average interarrival times (per subsystem), were run with each of six subjects. (The actual interarrival times were exponentially distributed about the mean.) To equally distribute the effects of practice obtained during a sequence of trials, a balanced design was employed, such that the three trials were given in a different order to each subject (see Table 1). Six subjects participated, all of which were male students or former students in engineering.

During this first part of the experiment, two sessions, three ten-minute trials in each, were run with each subject. The first session served to train the subjects for the monitoring task; little improvement in performance was noted during the second session. (Only the data from the second session was used for analysis.)

Data from this part of the experiment consisted of the following:

1. Time of occurrence of a subsystem event

2. Time of response to the event

3. Time of completion of diagnostic action for the event.

Failures to complete diagnostic action after an initial response (incorrect actions) were also noted, as well as responses to nonexistent events (false alarms).

The second part of the experiment consisted of both monitoring and control tasks. After considerable training, the subjects participated in four formal experimental sessions, each of which involved two trials of about 15 minutes duration. For the first trial of a session, the subject was given a simple map (few turns) to follow (Figure 3). In the second trial, the course was more complex (Figure 4). The mean event interarrival time for the monitoring task remained the same during each session, but varied from session to session, as indicated in Table 1.

To establish baseline performance for the control task, the first session included only the aircraft control, but no monitoring. The remaining sessions were run with the three levels of monitoring workload used earlier. The balanced design for varying the monitoring task workload was also used in these sessions (Table 1). Prior to performing the final trials, the subjects practiced "flying" the airplane simulator, both without and with monitoring tasks. When they were able to fly the complex map, with the lowest (30 sec.) subsystem event interarrival time, and also maintain control of aircraft attitude and position, the final trials were run.

Data obtained from this part of the experiment (sampled every 0.25 sec.) included the three measures listed above, as well as control task information:

4. Aircraft position and attitude

5. Pilot control inputs.

Performance data for four levels of monitoring workload and three levels of control task workload were thus obtained from the experiment as a whole.

RESULTS

The raw data obtained during the experiment was analyzed to obtain subsystem event service time and waiting time statistics, summarized in Table 2. Errors (false alarms and incorrect actions) were also counted, and are shown in Tables 3 and 4. Table 5 shows three measures of control task performance (RMS perpendicular distance from the course, RMS pitch and roll angles), for the subsystem event arrival rates and maps used in the experiment (also presented graphically in Figure 5.)

Subsystem event service time is measured from initiation of diagnostic action to the completion of the action, provided no other subsystem action intervenes (after an incorrect action, a different, higher-priority subsystem event could be serviced). Figure 6 shows the empirical service time statistics. The average service time appears to be independent of subsystem event arrival rate. As discussed in THE MODEL section, the preemption of subsystem service by the control task is reflected as an increase in the subsystem service time. This increase is apparently not a function of control task difficulty, but simply of whether or not the control task is present.

The subsystem event waiting time is measured from the time of occurrence of an event to the time at which diagnostic action for that event is completed. Waiting time is the sum of two terms: response time, the time from event occurrence to initiation of diagnostic action; and service time, from initiation to completion of diagnostic action. As shown in Figure 7, waiting time increases with subsystem event arrival rate. This is due to the queueing effect on the response time.

As expected, waiting time also increases when the control task is added. This is due in part to the increased service time when the control task is present. However, larger service time alone does not account for all of the increase in waiting time. The control task also affects the response time. This is the queueing effect of the control task "customers" on the subsystem service performance. This effect is also only a function of the mere presence of the control task, rather than the control task difficulty, as was the effect on service time.

False alarms (Table 3) tend to occur less frequently with increasing control task difficulty and event arrival rate. When the control task was present, few false alarms were made; many of those which occurred were probably simply typing mistakes made in response to real subsystem events (this could not be detected). False alarms wasted on the order of 1 sec. of time, and probably had insignificant effect on subsystem task performance.

Incorrect actions (Table 4) tended to increase with subsystem event arrival rate, but showed no consistent variation with control task difficulty. Incorrect actions are significant because, on the average, one-half the average service time is wasted. Furthermore, once an incorrect action is made, the pilot can switch to a higher-priority event, without completing the diagnostic already begun. This contributes unevenly to the waiting time across the subsystems; such an occurrence during service of subsystem 6, for example, might result in that diagnostic action being delayed considerably, while similar preemption during service of subsystem 2 would not result in as long a delay. The impact of the incorrect actions is thus to increase both the average and standard deviation of waiting time for lower-priority processes.

Each data item from the results of the simulation to be discussed here is based on 10,000 simulated events. As might be expected, for all three sets of simulation results, the computer simulation is more consistent (smaller standard deviation) than the experimental subjects.

Table 6 shows both the empirical subsystem waiting time and the results from simulation Model 1. Input parameters to simulation Model 1 were:

1. Probability of incorrect action, given that a subsystem event has occurred. This was approximated from the empirical data by dividing the number of incorrect actions by the number of events.

2. False alarm arrival rate and service rate. The service rate was assumed to be five times the average subsystem event service rate for the task, because only one of the five levels of the checklist need be scanned to realize a false alarm has been made. The arrival rate for false alarms was obtained by dividing the number of false alarms by the total time during which false alarms could occur. (The total time during which false alarms could occur was equal to the total elapsed time minus the total time spent servicing false alarms, since false alarms could not occur while one was being serviced.)

3. Subsystem event arrival rate, one of the independent variables in the experiment.

4. Average subsystem event service rate, obtained from the empirical service time data.

5. Erlang service time distribution shape parameter k, the square of the ratio of mean to standard deviation. This was calculated from the empirical service time data (Table 2). This value averaged 19 for the monitoring-only tasks, and was 3 for all other tasks.

A subsystem event could not be detected immediately. By averaging the minimum times from event occurrence to initial response which were recorded in the experimental trials, 4.5 sec. was estimated as the average time after occurrence of an event before it could be detected. In the computer simulation, events are detected immediately; to account for the detection time delay, 4.5 sec. is added in the simulation to the basic waiting time, to give the adjusted waiting time.

Summarizing the results from Model 1, when no control task is present, the model predictions match the empirical data well. Both mean and standard deviation of waiting times for individual subsystems are close. However, when the control task is present, a close match is obtained only for the high subsystem arrival rate condition. With lower arrival rates, the model predictions are smaller than the empirical results. This can be explained in the following way. With high arrival rate, events occur frequently and most control actions preempt subsystem diagnostics in progress. The control task effect is therefore almost entirely included in the increased service time for subsystem events. With lower arrival rates, a significant portion of the control actions can be performed without preempting subsystem diagnostics, but may increase the time to respond to a subsystem event. Thus, for the lower subsystem event arrival rates, the control task is not adequately modeled as simply an increase in service time for subsystem events.

Simulation Model 2, which uses an effective arrival rate (see THE MODEL) to account for the control task, produced a good match to empirical data (see Table 7). The simulation input parameters are the same as those above, except that an effective arrival rate of subsystem events $\lambda_{effective} = \lambda_{actual} + \Delta\lambda$ is used instead of the actual arrival rate. $\lambda_{effective}$ was obtained by varying $\Delta\lambda$ until a close fit to the empirical waiting time data was found. (Since Model 1 produced an adequate fit to empirical data for the high arrival rate conditions, the effective arrival rate was only computed for low and medium arrival rates, both simple and complex maps.) The various values of $\Delta\lambda$ were averaged (to give 0.0095), and this average used to calculate the effective arrival rates used to produce the results summarized in Table 7.

The results of simulation Model 3 are shown in Table 8. The probability of incorrect action, false alarm arrival rate and service rate, and subsystem event arrival rate are the same parameters as for Model 1. The subsystem event service rate was the average of the three rates obtained from the baseline monitoring-only tasks. The Erlang shape parameter k was the average of the three values obtained for the monitoring-only tasks.

The arrival rate of customers in the control task queue λ_c was calculated from the number of distinct control actions counted (N_c), the elapsed time of the corresponding trials (T_e), and the average duration of the control actions (T_d) (all empirical quantities) using:

$$\lambda_c = \frac{N_c}{T_e - N_c * T_d}$$

(Actually, only aileron control inputs were counted; elevator inputs were infrequent and of short duration compared to aileron inputs. See Figure 8 for an example of aileron control input vs. time for one trial.) The service rate for the control task queue was varied until close fits to empirical waiting time averages (across all subsystems) were obtained. The service rates for simple map were then averaged, as well as those for the complex map. An Erlang shape parameter k of 2 was used for the control task queue service time distribution. These values were used to produce the results shown in Table 8.

It is important to note that the results obtained for this third model were based on the assumption that control task service rate does not vary with subsystem event arrival rate. This assumption may be inappropriate. However, if control task service rate varies with both control task difficulty and subsystem event arrival rate, one could arbitrarily match almost any data. What is needed is a method of determining the control task service rate directly from the control signals. This is the most important extension needed by this queueing model.

SUMMARY AND CONCLUSIONS

The suitability of three alternative queueing models of the flight management situation was investigated, and the results presented. Attention was focused on predicting pilot performance in the subsystem monitoring task. In the first model, the subsystem monitoring task was explicitly modeled as a $M/E_k/1$:NPRP/N/N queueing system. The control portion of the task was included in the empirical subsystem event service time, which increased when the control task was present. This model adequately predicted the subsystem event waiting time performance only for the tasks with highest arrival rates.

The second model accounted for the control portion of the task (with low and medium arrival rates) by using an effective arrival rate for subsystem events. By adding an increment of arrival rate to the actual event arrival rate, the empirical waiting time performance could be predicted satisfactorily by computer simulation. However, using an effective arrival rate leaves much to be desired from the point of view of what is physically happening in the overall task.

The third model incorporated the control task as a separate, special queue which could preempt subsystem event service whenever a customer arrived. The task was thus modeled as a $M/E_k/1$:PRP/N/N queueing system. An estimate of the arrival rate for control task queue customers was made. The service rate for the control task was used as a fitting parameter. By varying this parameter in the simulation, a reasonable fit to empirical data was obtained.

The three models presented represent successive refinements in an attempt to incorporate the control task into the queueing model of the flight management situation. The subsystem monitoring task was easily formulated using a queueing description, and its basic representation was the same in all three model refinements. The third model refinement most closely approaches the real situation with respect to representation of the control and subsystem monitoring tasks; it is therefore intuitively appealing (although the control task parameters are not well-defined).

The goal of this research was to determine if a queueing model would provide an adequate description of pilot decision making in a multi-task, control and monitoring situation. Reasonable predictions of performance of the subsystem task were obtained with little or no fine tuning of parameters. This suggests that the model is promising, and can be successfully employed not only in the adaptive computer aiding scheme mentioned earlier, but perhaps in a broader class of similar multi-task situations.

For example, the proposed queueing model could be fit to multi-task situations and then the fractions of attention required by each task could easily be determined since they are inherent outputs of a queueing formulation. The advantage of this approach is that the fractions of attention are no longer free parameters as they are in control theory models of human decision making in multi-task situations [12, 13].

REFERENCES

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

2. W. B. Rouse, "Design of Man-Computer Interfaces for On-Line Interactive Systems," Proceedings of the IEEE, Special Issue on Interactive Computer Systems, Vol. 63, No. 6, pp. 847-857, June 1975

3. W. B. Rouse, "Human-Computer Interaction in Multi-Task Situations," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, No. 5, pp. 384-392, May 1977.

4. Y. Chu and W. B. Rouse, "Optimal Adaptive Allocation of Decision Making Responsibility Between Human and Computer in Multi-Task Situations," to appear in the Proceedings of the 1977 International Conference on Cybernetics and Society, September 1977.

5. J. H. Blakelock, Automatic Control of Aircraft and Missiles, New York: John Wiley & Sons, Inc., 1965.

6. R. S. Walden, "A Queueing Model of Pilot Decision Making in a Multi-Task Flight Management Situation," MSME thesis, University of Illinois at Urbana-Champaign, July 1977.

7. D. L. Kleinman, S. Baron, and W. H. Levison, "A Control Theoretic Approach to Manned-Vehicle Systems Analysis," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, pp. 824-832, December 1971.

8. J. W. Senders, "The Human Operator as a Monitor and Controller of Multidegree of Freedom Systems," IEEE Transactions on Human Factors in Electronics, Vol. HFE-5, No. 1, pp. 2-5, September 1964.

9. J. R. Carbonell, "A Queueing Model of Many-Instrument Visual Sampling," IEEE Transactions on Human Factors in Electronics, Vol. HFE-7, No. 4, pp. 157-164, December 1966.

10. Y. Chu, "Optimal Adaptive Allocation of Decision Making Responsibility Between Human and Computer in Multi-Task Situations," Ph.D. thesis in progress, University of Illinois at Urbana-Champaign.

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

12. R. E. Curry, D. L. Kleinman, and W. C. Hoffman, "A Design Procedure for Control/Display Systems," Human Factors, Vol. 19, No. 5, October 1977.

13. S. Baron and W. H. Levison, "Display Analysis with the Optimal Control Model of the Human Operator," Human Factors, Vol. 19, No. 5, October 1977.

Table 1. Experimental design

Subject	Session Number 1: No control task			2: Control Task, Simple Map	2: Control Task, Complex Map	3: Control Task, Simple Map	3: Control Task, Complex Map	4: Control Task, Simple Map	4: Control Task, Complex Map	5: Control Task, Simple Map	5: Control Task, Complex Map
1	30*	60	90			30	30	60	60	90	90
2	30	90	60			30	30	90	90	60	60
3	60	30	90	No		60	60	30	30	90	90
4	60	90	30	Monitoring		60	60	90	90	30	30
5	90	30	60			90	90	30	30	60	60
6	90	60	30			90	90	60	60	30	30

* Table entries are mean subsystem event interarrival times (sec.)

Table 2. Subsystem event service performance (empirical)

Subsystem No.	N (0.01111)	T_s	σ_s	T_w	σ_w	N (0.01667)	T_s	σ_s	T_w	σ_w	N (0.03333)	T_s	σ_s	T_w	σ_w
								No Map							
1	18	5.55	0.81	12.97	3.06	82	5.57	1.32	12.86	3.67	84	5.96	2.08	14.67	4.20
2	25	5.84	1.62	12.10	3.13	48	5.40	1.10	14.11	4.28	82	5.42	1.19	14.68	3.93
3	30	5.36	0.58	12.38	2.51	46	5.28	1.21	14.43	5.36	82	5.29	1.31	17.87	7.25
4	24	6.22	2.46	12.80	4.67	36	5.27	0.67	15.91	5.88	90	5.44	1.35	18.69	10.07
5	32	5.41	1.39	12.62	5.74	42	5.39	0.62	15.74	6.38	58	5.78	1.78	24.26	18.38
6	48	5.58	0.91	13.55	3.82	35	5.37	0.90	17.00	7.56	52	5.39	1.25	32.88	27.59
All*	177	5.63	1.38	12.82	4.01	289	5.41	1.07	14.62	5.48	448	5.55	1.54	19.42	14.10
								Simple Map							
1	34	8.59	6.53	19.17	6.79	44	7.92	3.87	19.69	7.26	56	8.56	4.59	20.85	6.95
2	43	8.88	6.23	18.98	8.69	53	8.49	5.22	20.56	9.75	79	9.60	7.04	23.42	9.58
3	18	7.51	4.95	23.93	8.46	54	10.01	7.59	24.91	12.35	79	9.74	5.94	27.81	12.34
4	54	9.57	5.80	24.87	12.50	51	8.74	5.20	28.21	18.15	67	8.45	4.69	32.68	21.43
5	39	9.54	6.87	31.43	24.63	52	8.33	4.23	35.97	37.04	42	9.82	5.83	54.08	40.45
6	31	10.27	6.95	34.18	26.39	34	8.55	4.43	55.67	55.40	30	7.03	2.13	109.39	151.78
All	219	9.21	6.28	25.24	17.28	288	8.71	5.33	29.52	28.81	353	9.06	5.59	36.71	53.13
								Complex Map							
1	40	10.87	7.47	22.21	8.57	41	9.37	5.43	22.78	10.24	54	8.45	4.94	21.75	10.39
2	41	9.57	5.29	20.74	8.19	49	10.93	6.94	22.46	9.47	86	9.32	6.88	22.58	9.37
3	35	8.39	4.36	25.10	14.80	52	9.28	6.56	27.03	17.10	74	8.97	5.61	26.89	14.33
4	46	8.70	5.49	24.10	12.63	53	7.90	3.44	29.94	21.15	71	8.81	5.17	36.82	29.11
5	52	9.48	7.61	32.54	27.72	47	9.94	6.02	39.06	40.22	56	8.80	5.08	50.19	56.27
6	18	9.52	6.15	47.80	34.61	35	9.71	5.50	52.76	91.51	30	8.09	3.29	86.50	66.71
All	232	9.42	6.25	27.06	19.97	277	9.49	5.79	31.44	39.59	381	8.84	5.50	35.18	37.00

Subsystem event arrival rate (events/sec): 0.01111, 0.01667, 0.03333

* Cumulative statistic, across all six subsystems
N - Number of subsystems events recorded
T_s, σ_s - Event service time average, standard deviation (sec.)
T_w, σ_w - Event waiting time average, standard deviation (sec.)

	Subsystem event arrival rate (events/sec.)		
	0.01111	0.01667	0.03333
No map	51*	20	26
Simple map	9	6	6
Complex map	4	6	13

* Table entries are total number of false alarms of six subjects for the specified task.

Table 3. False alarms

	Subsystem event arrival rate (events/sec.)		
	0.01111	0.01667	0.03333
No map	17*	24	50
Simple map	26	19	39
Complex map	15	21	38

* Table entries are total number of incorrect actions of six subjects for the specified task.

Table 4. Incorrect actions

Table 5. Control task performance (empirical)

Statistic	Subsystem event arrival rate (events/second)			
	0	0.01111	0.01667	0.03333
		Simple Map		
Mean position error (ft.)	1489	1852	3847	2214
RMS position error (ft.)	1961	2391	6135	2698
Mean course error (ft.)	790	1074	1096	1186
RMS course error (ft.)	1268	1609	1732	1730
RMS altitude error (ft.)	2700	4616	4395	6694
RMS pitch angle (radians)	0.096	0.118	0.134	0.119
RMS roll angle (radians)	0.294	0.304	0.329	0.318
		Complex Map		
Mean position error (ft.)	3958	3247	3443	5203
RMS position error (ft.)	6526	4081	4309	6731
Mean course error (ft.)	1626	1638	1467	1921
RMS course error (ft.)	2334	2401	2033	2611
RMS altitude error (ft.)	6694	3842	3402	5355
RMS pitch angle (radians)	0.158	0.147	0.136	0.171
RMS roll angle (radians)	0.427	0.428	0.415	0.406

Table 6. Comparison of empirical data and queueing Model 1 simulation results

Subsystem No.	Subsystem event arrival rate (events/sec.) 0.01111				0.01667				0.03333			
	T_{wm}	σ_{wm}	T_{we}	σ_{we}	T_{wm}	σ_{wm}	T_{we}	σ_{we}	T_{wm}	σ_{wm}	T_{we}	σ_{we}
					No Map							
1	11.86	2.51	12.97	3.06	11.88	2.33	12.86	3.67	13.28	2.97	14.67	4.20
2	12.16	2.92	12.10	3.13	12.19	2.68	14.11	4.28	14.14	3.82	14.68	3.93
3	12.42	3.39	12.38	2.51	12.50	3.21	14.43	5.36	15.69	5.72	17.87	7.25
4	12.72	3.65	12.80	4.67	13.30	4.27	15.91	5.88	18.27	8.86	18.69	10.07
5	13.32	4.63	12.62	5.74	14.02	5.87	15.74	6.38	22.22	14.03	24.26	18.38
6	13.79	5.38	13.55	3.82	14.91	6.86	17.00	7.56	29.54	25.83	32.88	27.59
All*	12.71	3.92	12.82	4.01	13.10	4.58	14.62	5.48	18.21	13.03	19.42	14.10
					Simple Map							
1	17.91	7.23	19.17	6.79	17.58	7.15	19.69	7.26	20.47	7.56	20.85	6.95
2	18.47	8.41	18.98	8.69	18.47	7.79	20.56	9.75	22.87	9.60	23.42	9.58
3	19.11	9.17	23.93	8.46	19.41	9.64	24.91	12.35	27.11	14.25	27.81	12.34
4	20.39	11.74	24.87	12.50	21.95	12.08	28.21	18.15	36.38	24.30	32.68	21.43
5	21.67	13.99	31.43	24.63	24.25	16.31	35.97	37.04	54.51	46.07	54.08	40.45
6	24.46	18.37	34.18	26.39	28.90	23.29	55.67	55.40	102.03	109.94	109.39	151.78
All	20.29	12.22	25.24	17.28	21.60	14.16	29.52	28.81	35.81	43.68	36.71	53.13
					Complex Map							
1	17.49	7.50	22.21	8.57	19.14	7.71	22.78	10.24	19.91	7.46	21.75	10.39
2	18.46	8.07	20.74	8.19	20.33	8.74	22.46	9.47	22.39	9.69	22.58	9.37
3	19.28	9.65	25.10	14.80	22.27	11.29	27.03	17.10	26.54	13.39	26.89	14.33
4	19.94	10.84	24.10	12.63	25.02	14.79	29.94	21.15	35.03	22.92	36.82	29.11
5	21.47	13.16	32.54	27.72	29.34	21.50	39.06	40.22	51.69	43.92	50.19	56.27
6	23.23	16.61	47.80	34.61	35.64	31.04	52.76	91.51	94.50	97.10	88.50	66.71
All	19.92	11.46	27.06	19.97	24.90	18.07	31.44	39.59	34.54	40.26	35.18	37.00

* Cumulative statistic, across all six subsystems

T_{wm}, σ_{wm} - Model 1 event waiting time average, standard deviation (sec.)

T_{we}, σ_{we} - Empirical event waiting time average, standard deviation (sec.)

Table 7. Comparison of empirical data and queueing Model 2 simulation results

Subsystem No.	Subsystem event effective arrival rate (events/sec.) ($\lambda_{effective} = \lambda_{actual} \bullet 0.0095$) 0.0206				0.0262			
	T_{wm}	σ_{wm}	T_{we}	σ_{we}	T_{wm}	σ_{wm}	T_{we}	σ_{we}
				Simple Map				
1	19.63	7.89	19.17	6.79	18.89	7.12	19.69	7.26
2	21.09	9.26	18.98	8.69	21.02	8.72	20.56	9.75
3	23.12	11.97	23.93	8.46	23.40	11.76	24.91	12.35
4	27.39	17.02	24.87	12.50	28.17	17.78	28.21	18.15
5	34.90	26.56	31.43	24.63	36.97	30.10	35.97	37.04
6	46.07	43.47	34.18	26.39	53.04	49.79	55.67	55.40
All*	27.59	23.01	25.24	17.28	28.39	25.45	29.52	28.81
				Complex Map				
1	19.80	7.90	22.21	8.57	20.26	7.60	22.78	10.24
2	21.83	9.61	20.74	8.19	22.77	10.01	22.46	9.47
3	24.19	12.29	25.10	14.80	25.96	12.97	27.03	17.10
4	27.83	17.37	24.10	12.63	32.92	21.51	29.94	21.15
5	34.42	24.97	32.54	27.72	43.34	36.55	39.06	40.22
6	45.38	42.74	47.80	34.61	67.87	69.66	52.76	91.51
All	27.95	22.71	27.06	19.97	32.33	32.58	31.44	39.59

* Cumulative statistic, across all six subsystems

T_{wm}, σ_{wm} - Model 2 event waiting time average, standard deviation (sec.)

T_{we}, σ_{we} - Empirical event waiting time average, standard deviation (sec.)

$\lambda_{effective}$ - Effective subsystem event arrival rate used in simulation

λ_{actual} - Actual subsystem event arrival rate used in experiment

Table 8. Comparison of empirical data and queueing Model 3 simulation results

Subsystem event arrival rate (events/sec.) (μ_s = 0.181, k_s = 19 for all tasks.)

Subsystem No.	0.01111				0.01667				0.03333			
Simple Map (λ_c = 0.15, μ_c = 0.20, k_c = 2)												
	T_{wm}	σ_{wm}	T_{we}	σ_{we}	T_{wm}	σ_{wm}	T_{we}	σ_{we}	T_{wm}	σ_{wm}	T_{we}	σ_{we}
1	19.59	8.54	19.17	6.79	20.35	8.75	19.59	7.26	21.74	8.84	20.85	6.95
2	20.81	9.65	18.98	8.69	22.19	10.35	20.56	9.75	24.90	11.40	23.42	9.58
3	22.21	11.32	23.93	8.46	23.99	12.58	24.91	12.35	30.68	16.20	27.81	12.34
4	24.43	13.99	24.87	12.50	27.57	17.06	28.21	18.15	43.05	28.99	32.68	21.43
5	26.52	17.04	31.43	24.63	32.34	23.53	35.97	37.04	70.40	60.58	54.08	40.45
6	29.76	22.65	34.18	26.39	40.75	35.58	55.67	55.40	138.07	139.16	109.39	151.78
All*	23.74	14.86	25.24	17.28	27.33	20.59	29.52	28.81	41.21	53.79	36.71	53.13
Complex Map (λ_c = 0.14, μ_c = 0.18, k_c = 2)												
1	20.27	9.21	22.21	8.57	21.20	9.45	22.76	10.24	22.05	9.48	21.75	10.39
2	20.88	10.02	20.74	8.19	22.82	10.82	22.46	9.47	25.21	11.85	22.58	9.37
3	22.37	11.75	25.10	14.80	25.05	13.39	27.03	17.10	30.75	16.88	26.89	14.33
4	23.21	13.26	24.10	12.63	29.76	19.09	29.94	21.15	42.80	31.38	36.82	29.11
5	26.14	16.88	32.54	27.72	34.38	26.04	39.05	40.22	70.36	60.35	50.19	56.27
6	29.33	21.63	47.80	34.61	43.33	37.52	52.75	91.51	141.03	152.17	88.50	66.71
All	23.64	14.69	27.06	19.97	28.85	22.26	31.44	39.59	41.53	56.25	35.18	37.00

* Cumulative statistic, across all six subsystems

T_{wm}, σ_{wm} - Model 3 event waiting time average, standard deviation (sec.)

T_{we}, σ_{we} - Empirical event waiting time average, standard deviation (sec.)

λ_c, μ_c, k_c - Control task customer arrival rate and service rate (customers/sec.), and Erlang distribution shape parameter

μ_s, k_s - Subsystem event service rate (events/sec.), and Erlang distribution shape parameter

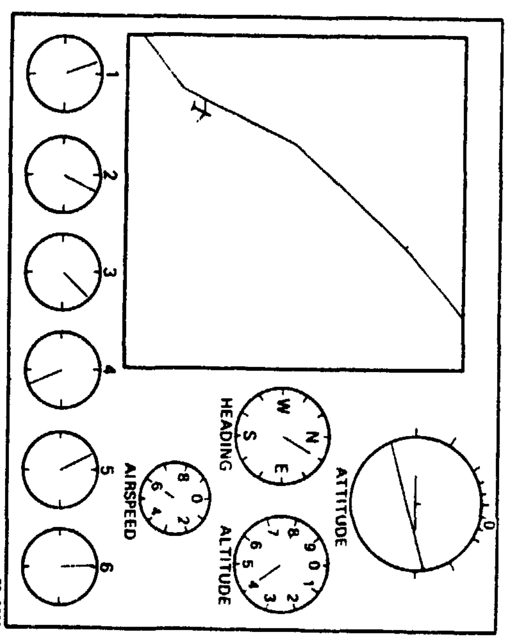

Figure 1. CRT display showing simulated cockpit instruments

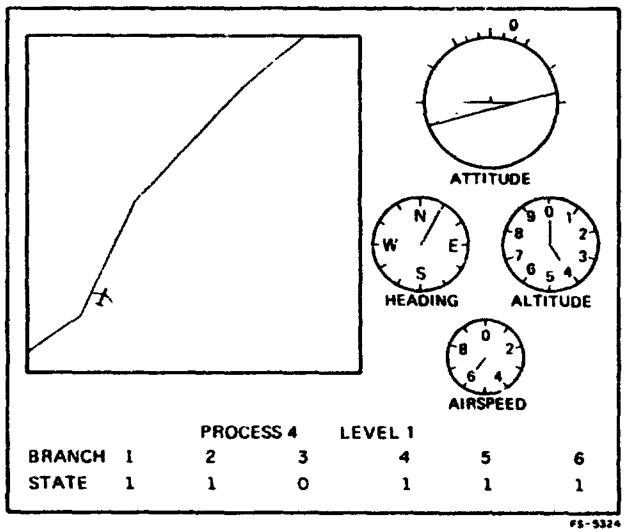

Figure 2. CRT display after pilot response to subsystem event

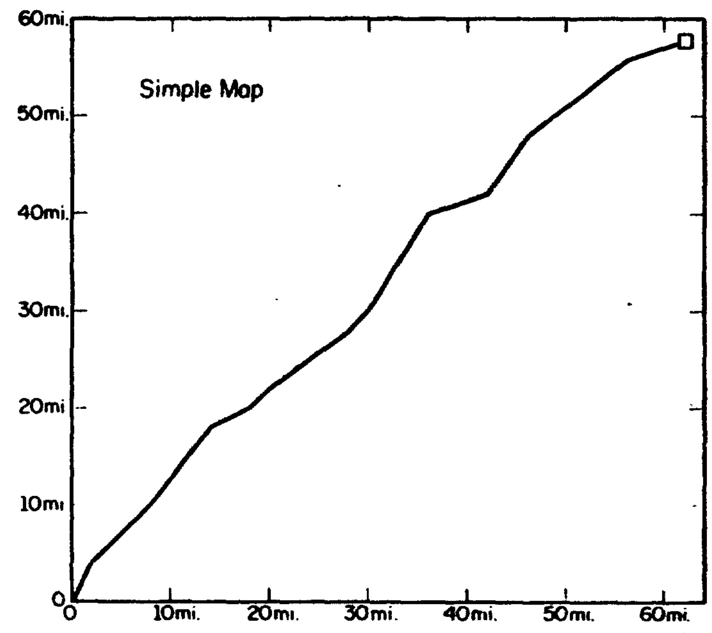

Figure 3. Simple map

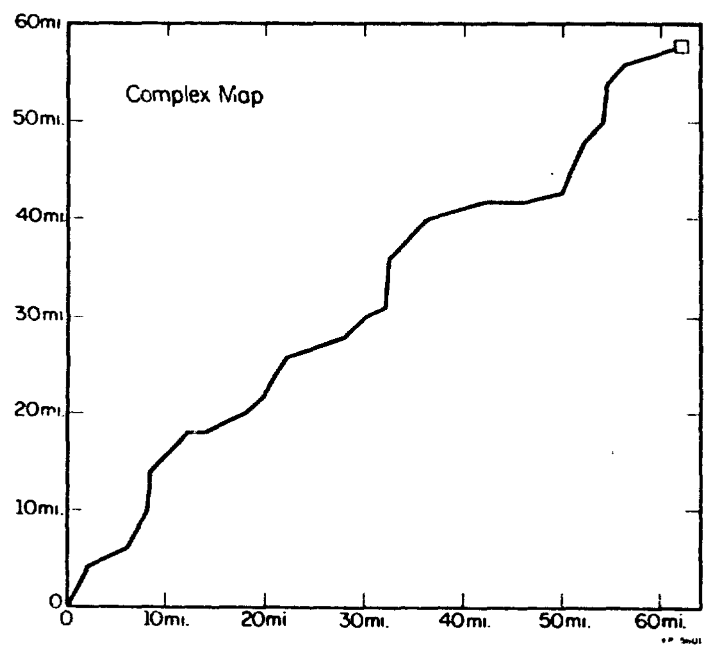

Figure 4. Complex map

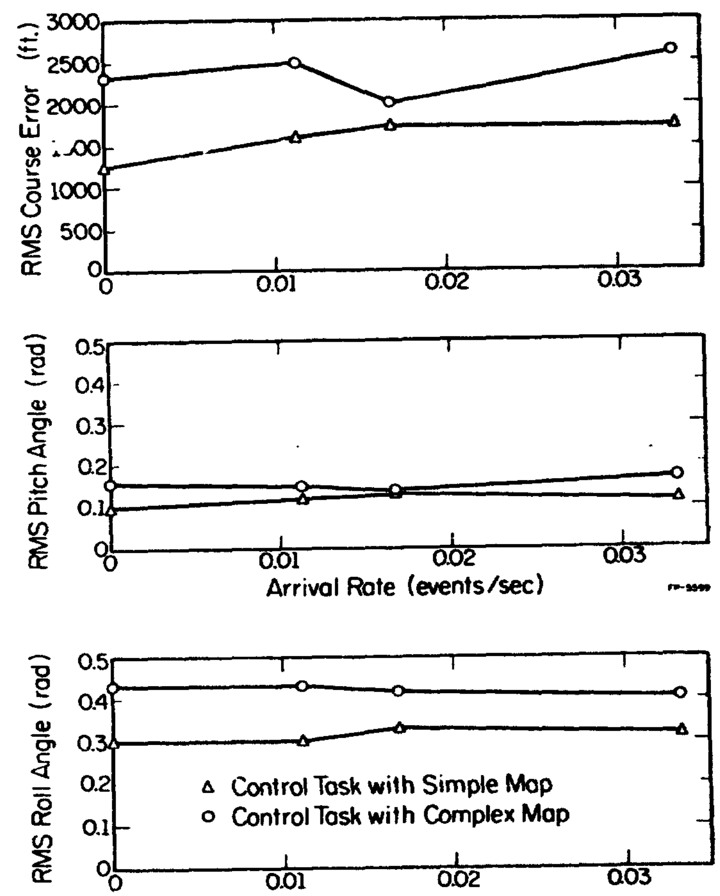

Figure 5. Control task performance

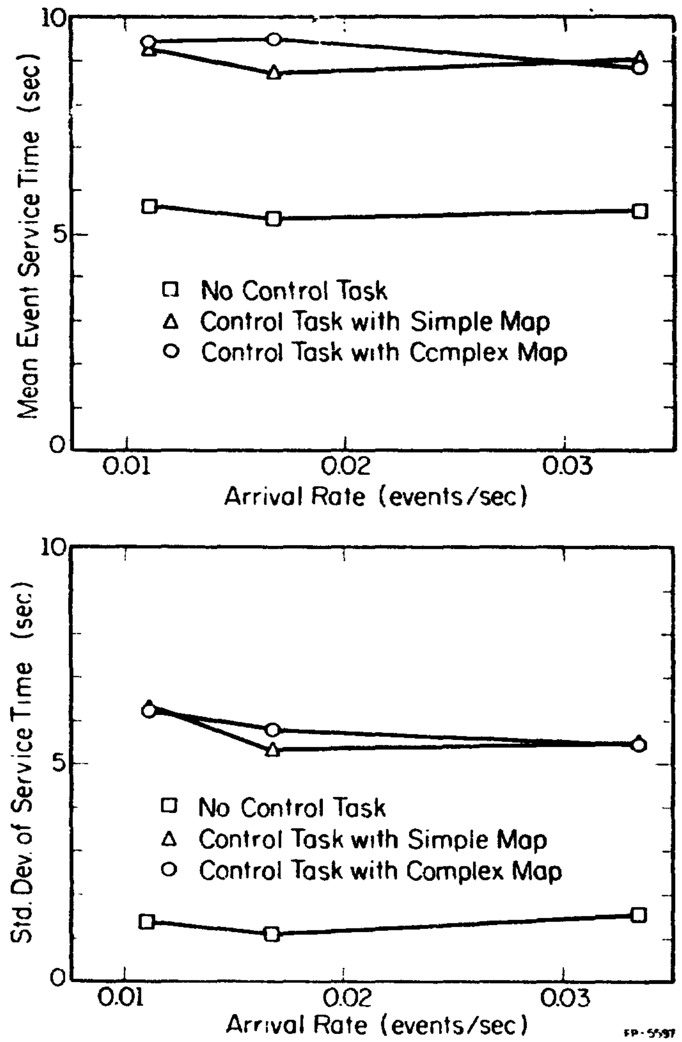

Figure 6. Subsystem event service time statistics (empirical)

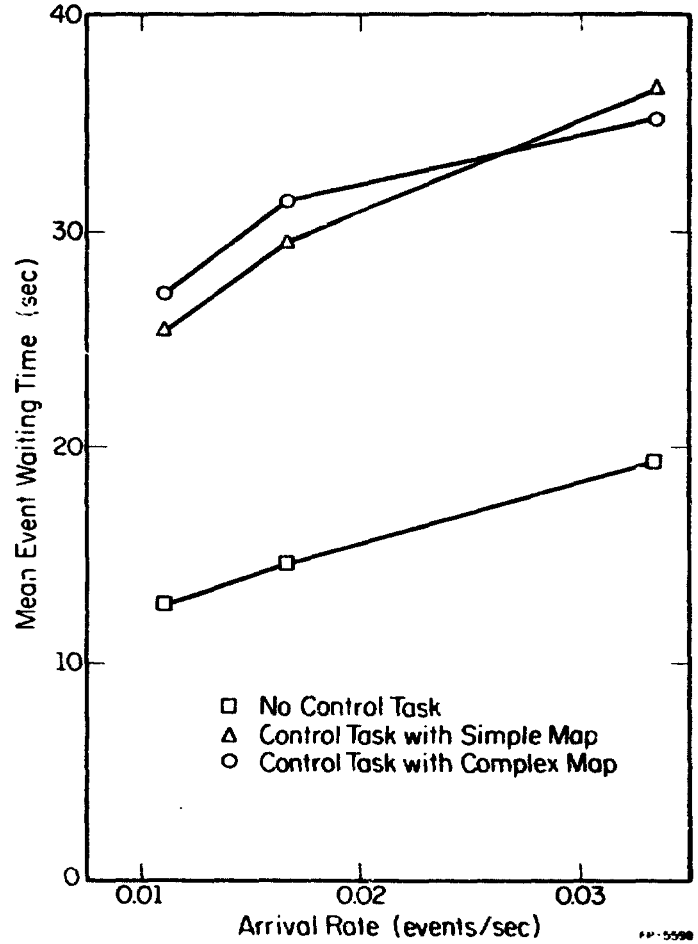

Figure 7. Average subsystem event waiting time (empirical)

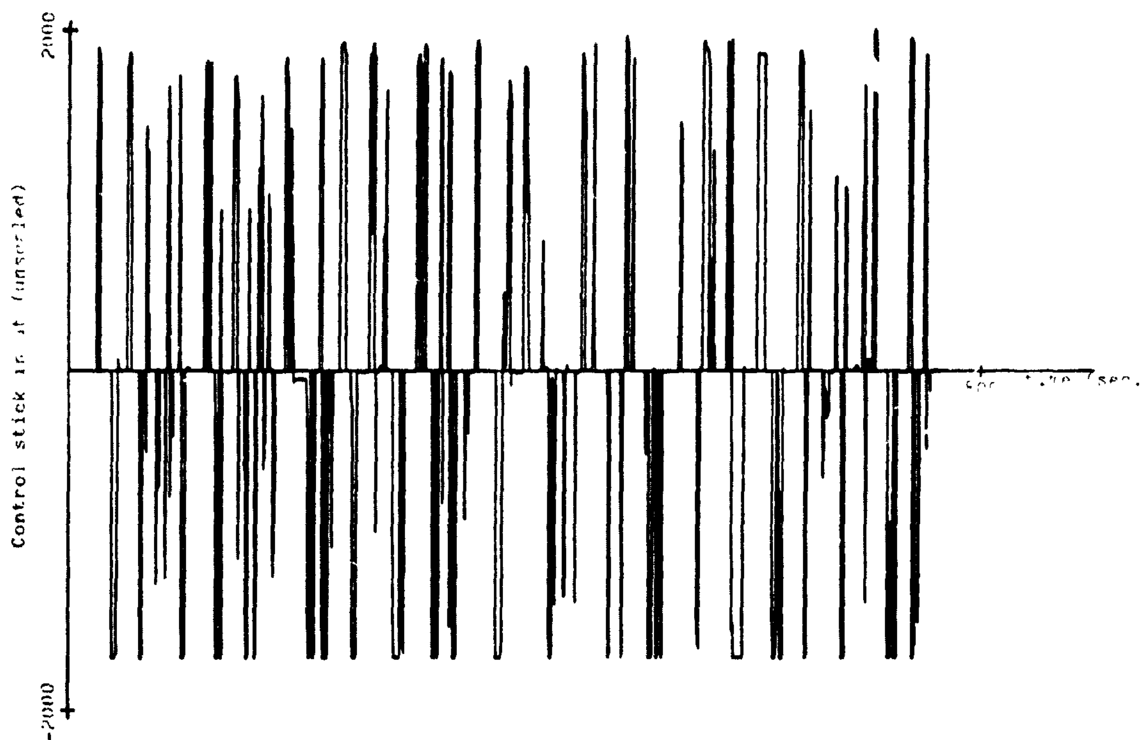

Figure 8. Aileron control stick input