DOCUMENT RESUME

ED 031 961 EM 007 419

By-Morfield, M. A.; And Others
Initial Experiments on the Effects of System Delay on On-Line Problem-Solving.
Massachusetts Inst. of Tech., Lexington. Lincoln Lab.
Spons Agency-Air Force Electronic System Div.
Report No-TN-1969-5-ESD-TR-69-158
Pub Date 24 Jun 69
Note-65p.
EDRS Price MF-$0.50 HC-$3.35
Descriptors-Analysis of Variance, Computer Programs, Display Panels, Graphs, Information Systems, *Input Output, Input Output Analysis, Interaction, *Interaction Process Analysis, *Problem Solving, *Program Design, Programing Problems, Systems Analysis, Tables (Data), Task Performance

The main purpose of the research reported in this document was to discover whether controlled experiments can be conducted on the relations between people and the complex computing systems which they use. Three increasingly complex experiments were designed to test the effect of varying delays of computer response on the number of commands issues per minute, as well as the total time needed to complete a task. The system used was a time-shared, on-line TX-2 computer and the Lincoln Reckoner, a subset of the programs in the executive system known as APEX. It was hoped that the experiments would not only further the knowledge of how people solve problems, but also aid in the design of new systems. The results indicate not only the feasibility of testing man-computer interaction, but also demonstrate more clearly the differences between the subjects' behavior in the various tasks in such indices of performance as net completion time and number of outputs. In addition, the gross completion time curves and output rate curves indicate that experiments large enough to produce stable curves would be feasible. Appendices, a reference list, diagrams, charts, tables and graphs are included in the document. (SH)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

INITIAL EXPERIMENTS ON THE EFFECTS OF SYSTEM DELAY ON ON-LINE PROBLEM-SOLVING

M. A. MORFIELD
R. A. WIESEN
M. GROSSBERG
D. B. YNTEMA

Group 25

TECHNICAL NOTE 1969-5

24 JUNE 1969

LEXINGTON MASSACHUSETTS

ABSTRACT

Three experiments explored the way in which delay in the response of the system affects the user's performance in solving problems with an on-line computing service. Each experiment was more ambitious than the preceding: the subject's task was more realistic and more complex. In each experiment there were four subjects under delay conditions of about 1 sec. to 100 sec. The on-line computing service was the Lincoln Reckoner.

As expected, the average time the user required to complete a task increased as the response-delay increased, and the rate at which he demanded service declined as the delay increased. The relation of net completion time (time to complete the task, minus the time during which the user was waiting for a response) to response delay depended on the type of task. In the more realistic experiments, the net completion time increased with delay (suggesting that long delays are distracting). The number of outputs (i. e. , displays or type-outs) per task was also considered.

The main conclusion is that controlled experiments of this kind are feasible and can be used as the basis for design of on-line computing services.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

CONTENTS

INTRODUCTION

This note presents three experiments on the effects of response-delays in a time-shared, on-line computing system. The main purpose was to discover whether controlled experiments can be conducted on the relations between people and the complex computing systems with which they work. We hoped to find out whether such human factors experiments, necessarily based on the complexity of interactions between men and systems, are feasible. If they are, the results should be useful both in the design of computing systems and in achieving a better understanding of how people solve problems.

It is important to note that these experiments investigated the use of the computer as a computing device rather than as a programming device. It is not at all necessary to know how to "program," in the sense of compiler language or machine language, in order to solve substantive problems with a "problem-oriented" system of the kind used in these experiments. Human factors experiments on the performance of programmers have been done (1,2,3), and they suggest that further experiments of that kind would be useful. The previous experiments, however, have been primarily concerned with the differences between on-line and "batch" systems. The experiments reported in this note are, in contrast, concerned with substantive users instead of programmers, and they are attempts to make a parametric study of system effects. That is, we are concerned with establishing functional relations rather than just looking for significant differences between conditions.

There have traditionally been two points of view on the amount of delay acceptable in an on-line computer: one, that delay should be imperceptible, and that anything less ambitious is unacceptable in a really useful facility; the other, that the user is relatively unimportant compared with the machine, and that the time-sharing algorithm should therefore be designed to attain efficient machine performance, letting the delays fall where they may. Neither of these views really faces the question of designing a system for people to use in an organization that is concerned with costs. To be realistic, the designer must consider the trade-off between the cost of the computer and the time of the people who use it. One of the purposes of the present note is to show how empirical curves can be obtained to give the designer the data on user performance that he needs if he wants to design for minimum total cost.

We conducted three experiments, all on the effects of response-delay, differing primarily in the type of task the subjects performed. Each experiment was designed to have greater face validity than the preceding one, and thus introduced greater complexity into the subjects' task. One index of this complexity was the successively larger repertoire of computational tools needed to deal with the three kinds of tasks. In the first experiment the subjects needed only one routine, in the second they had about a dozen available, and in the third they could use almost the complete Lincoln Reckoner, a library of about 80 routines.

GENERAL PROCEDURE

Computation [illegible]ues Used

The three experiments were performed on the TX-2 computer, an experimental machine at the Lincoln Laboratory. The computer operated under a time-sharing executive system known as APEX. (4) The Lincoln Reckoner (5), a sub-set of the programs in the APEX public library, was the facility the subjects used to work on the tasks.

The Lincoln Reckoner has been described as "a time-shared system for on-line use in scientific and engineering research... it was designed... to find out what features of such a service will have an important effect on the amount of work the user gets done.. it offers a library of routines that concentrate on one particular application, numerical computations on arrays of data. It is intended for use in feeling one's way through the reduction of data from a laboratory experiment, or in trying out theoretical computations of moderate size." (5, p433)

Three basic features of a user-oriented system of this type are (5, pp437-8):

1. Automatic application of routines. Almost all of the clerical work needed to perform an operation — i. e., to apply a public routine — is done automatically. The system takes care of the location of the data, the dimensions of arrays and so forth. Ideally, all the user has to do is somehow indicate the operation he wants to apply, the data to which he wants it applied, and — perhaps — the way in which he will

identify the results when he wants to use them again . . .

2. Automatic retention of results in such a form that they can be used as operands for other routines. The results of operations are stored in such a fashion that they can be used later as inputs to other operations — including operations that the user did not have in mind when the results were obtained. He need only specify the name by which he wants to identify a result; the system remembers where the result has been stored and automatically records the descriptive information that will be needed if the result is to be used later as an operand — e.g., the dimensions are recorded if the result is an array of numbers.

3. Facilities for concatenation of routines. The user can define a sequence of operations and then use the sequence as he would use one of the primitive routines in the library. The new operation can be used as part of another sequence, and so on.

Subjects had direct access to the computer by way of a terminal consisting of a Lincoln Writer keyboard and printer, and a CRT display. Across the room there was a Xerox high-speed printer that provided on-line hard-copy. Textual information and graphs were presented on the CRT or the Xerox at the user's request. The graph-plotting automatically provided appropriate scales for axes, and would plot

up to three sets of x - y values.

The Independent Variable: Delay in Output

The independent variable in all three experiments was the amount of delay in each output from the computer. "Output," as used here, means that the subject has requested some response, such as a type-out or a CRT display, or that an error message has been typed by the machine because he violated a syntactic or semantic rule of the system. It should be remembered throughout this paper that the Reckoner, unlike many on-line systems, does not reply to every line of typing. Commands are saved and executed when the system has time; an "output" is produced only when the system gets to the execution of a command that requests an output, or a command that is in error.

There were five experimental conditions: the nominal delay in each output was 1, 3, 10, 30, or 100 sec. All five conditions were used in the first and second experiments, but only four conditions, all but the 3 sec. delay, were used in the third experiment. In the conditions in which the nominal delay was 3, 10, 30 or 100 sec., the actual delay of any one output varied within plus or minus 10% of the nominal value. In the condition in which the nominal delay was 1 sec., the machine was actually responding as quickly as possible; that is, the actual delay was simply the time required to do the computation and prepare the output, plus occasionally,

some extra delay because the machine was being time-shared among the four subjects.[1]

Outputs were trapped to a program that decided the extent of the delay. The program first selected a delay at random from a table that was produced each time the subject started a new task. The program then compared the time of the previous output with the time of the carriage return that led to the current output (each command is ended by a carriage return): the more recent of those two events was regarded as the start of the delay interval. If the delay interval was already greater than the selected delay by the time the trapping program gained control, the output was begun immediately. If, however, the selected delay was greater, the program waited until the delay interval became equal to the selected delay. The time of each carriage return, the selected delay, and the actual delay were recorded by the computer.

The Dependent Variables

The designer of an on-line computing service presumably would be interested in balancing the cost of the user's time against the cost of the computer system, and thus would want to know how the time that the user needs to complete his task

---

1. In the first experiment the mean delay in this condition was approximately 0.7 sec., in the second it was approximately 0.4 sec., and in the third it was, we judge, somewhere in between.

depends on the delay in the machine's response. In each of the three experiments we shall therefore present graphs showing how the time required to complete a typical task varies as a function of delay.

Since the delay in the machine's response may well affect the rate at which the user requests service, the designer would like to know how the number of commands issued per minute varies from one delay condition to another. In each experiment we shall present graphs of number of outputs per minute, although in the third experiment the use of output rate as a measure of the load the user puts on the system is tenuous. In that experiment, in contrast to the first two, the number of commands the subject gives between commands that produce outputs may very considerably.[2]

The measures we have just considered – completion time and output rate – are of interest to the system designer, but to understand the subject's behavior we need measures that reflect his behavior more directly. In each experiment we shall therefore present graphs of the number of outputs for a typical task and the net completion time for a typical task. The net completion time is defined as $T - N \times D$, where T is the actual time the subject required to complete the task, N is the number of outputs he received, and D is the nominal delay in the condition under which he was working. Thus the net completion time is approximately the time

---

2. It might be preferable to present the number of commands per unit time, but those data are not available yet.

required to complete the task, when the intervals during which the subject was waiting for an output are ignored.

Subjects

The four subjects of these experiments were the four authors of this note. Subjects knew the delay condition that was in effect on each task and acted under a set to finish each task as soon as possible. Subjects were free to take breaks between tasks. They worked for approximately two hours, excluding breaks, one evening a week, and each experiment required several weekly sessions.

Since the subjects were also the experimenters they spent a considerable amount of time exploring the three task types before experimentation formally began. For this reason the role of practice effects (very small) may be misleading in the data analyses of the experiments.

EXPERIMENT I: Railroad Track Tasks

Procedure

The first experiment, called the Railroad Track Experiment (RR), may be regarded as a very simple kind of problem-solving. Since it was the first experiment, we tried to choose a task that would be simple, but would require a large amount of interaction with the computer.

At the beginning of each RR problem the subject typed a message that showed he was ready to begin, and the machine replied by displaying on the CRT a 5" × 5" graph of a pair of parallel curves (like railroad tracks) separated by approximately $\frac{1}{4}$ ", and a horizontal line across the middle of the CRT. An example is shown in Fig. 1. The subject's task was to manipulate the horizontal line until it fell between the pair of curved lines. The tool with which he manipulated the line was a command that altered the line by adding to it a "bump," shaped like a Gaussian bell-curve, whose height, width, and horizontal location he specified. (Appendix A presents the details of the "bump" routine and how the subject specified its parameters.) Each successive command cumulated with the previous ones, and automatically displayed the altered line superimposed on the railroad tracks, which remained fixed. The old display remained until the altered display appeared. (The time until the altered display appeared was, of course, the independent variable.) When the manipulated line fell completely between the railroad tracks the time and a message "DONE" were typed, indicating that the problem was finished.

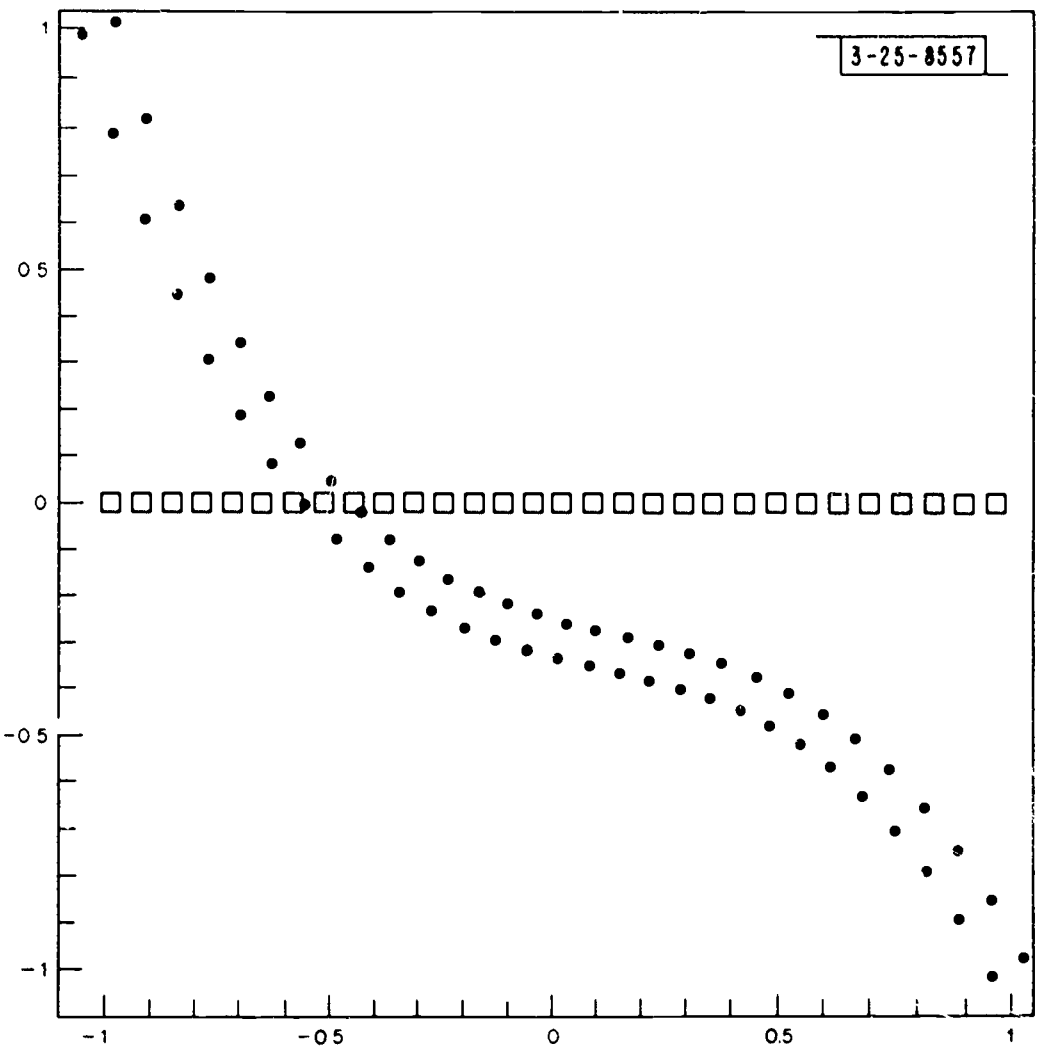


Fig. 1. An example of the initial display in the RR task.

Each RR problem was selected independently by a random process that specified the locus of the center of the parallel curves. (Appendix A presents further details.) Each subject did 25 tasks: the order of the five delay conditions (1, 3, 10, 30, and 100 sec.) was randomized over the 25 problems, with the restriction that the subject did five tasks under each condition.

Results

The experiment will be analyzed as a 4×5 factorial design (4 subjects and 5 conditions of delay) with five trials per cell — 100 trials in all. Completion time was the interval between the appearance of the display that stated the task, and the appearance of the "DONE" that marked its completion. In this experiment every command produced an output (an error message or new display), thus the number of outputs was simply the number of commands the subject gave, excluding the command that showed he was ready to begin, but including the command that produced the message "DONE."

Completion time and output rate. - The arithmetic means of the times required to complete the five tasks performed under each condition of delay are shown in Fig. 2 for each subject.[3] The number of outputs per minute under each delay

3. The means plotted in Fig. 2 are arithmetic, not geometric means. The geometric mean might be a more stable statistic, but it is not what the designer of a computer system needs. Given the arithmetic mean of the times required to complete the tasks in some population, he can make an accurate estimate of the total time that will be required to complete, say 1000 tasks drawn at random from that population: he simply multiplies the mean by 1000. But given the geometric mean, there is no good

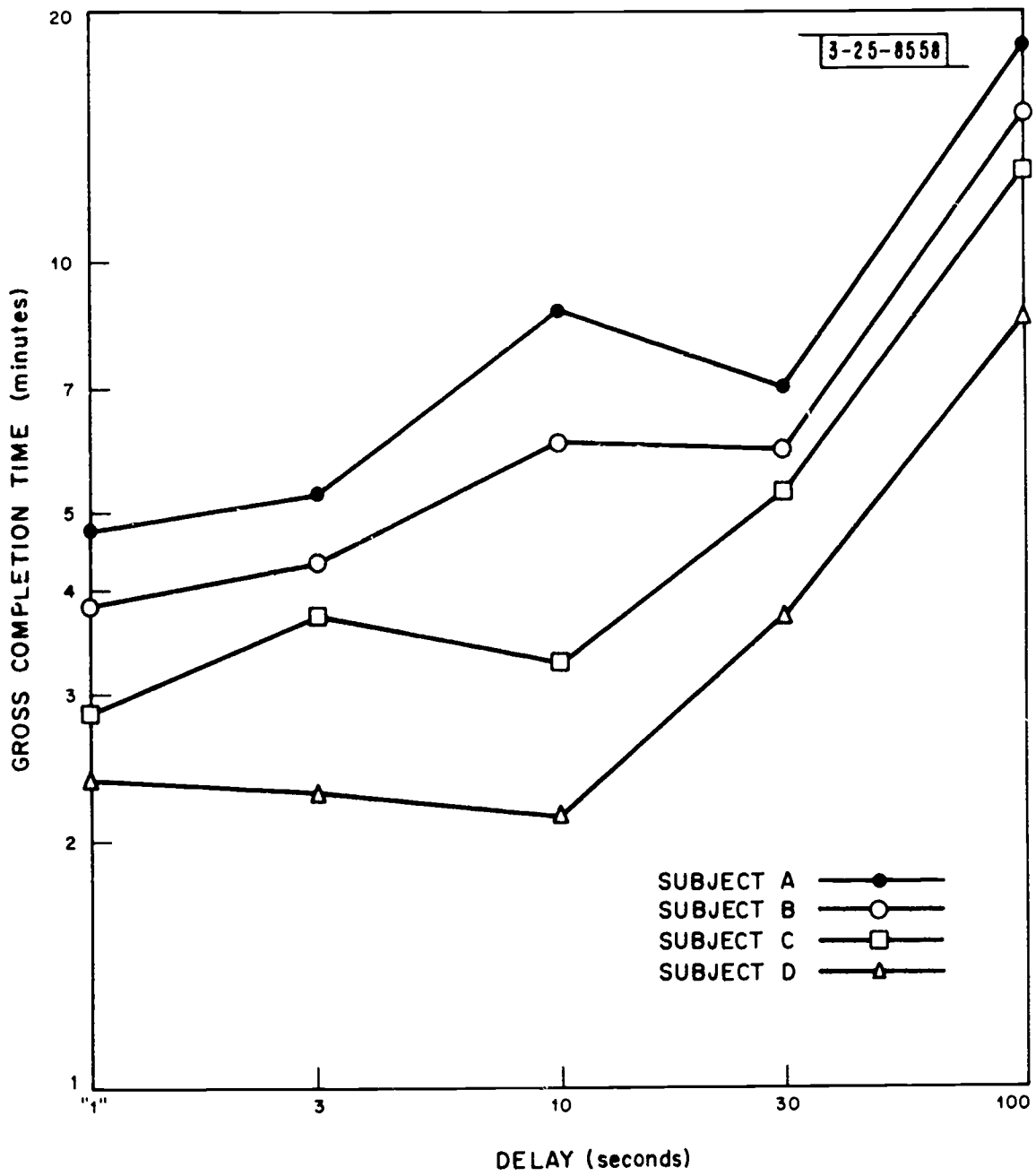


Fig. 2. Expenditure of the user's time: Arithmetic mean of time to complete a task in the RR experiment. (Log scales on both axes.)

condition are shown in Fig. 3 for each subject.[4] The curves are more ragged than we would wish if they were to be used in making decisions about the design of a real system, but a function is beginning to emerge. We conclude that if the RR task were of practical interest, it would be quite feasible to do an experiment large enough to trace out a function smooth enough for actual use.

Net completion time. - Table I presents an analysis of variance in which the dependent variable is the logarithm of net completion time (as defined in the section on General Procedure).[5] Subjects are treated as a random factor and delays as fixed. The main effects of subjects and delays were both significant, but the subject - by-delay interaction was not.[6]

---

way to estimate the total time: further information about the distribution of the population of completion times is needed.

4. Again the data have been combined in the way that is appropriate for presentation to a system designer. For each subject in each condition of delay, the total number of outputs received in the course of all five tasks was divided by the total time to complete those five tasks, and the quotient is the ordinate of the point plotted in Fig. 3. This way of computing output rates is appropriate because it weights units of time equally, rather than weighting trials equally.

5. In all of the analysis of variance tables in this paper, and in the linear regression analyses, the logarithmic transform has been applied to the raw data in order to improve the distribution of residuals. When the logarithmic transform is used, the geometric mean is plotted in the figures.

6. An examination of practice effects shows that the residual fell 0.0628 log units in the 25 trials, thus accounting for 1.3% of the error sum of squares. The error sum of squares in Table I is therefore slightly inflated, but since that inflation is only 1.3% an analysis of covariance does not seem worthwhile.

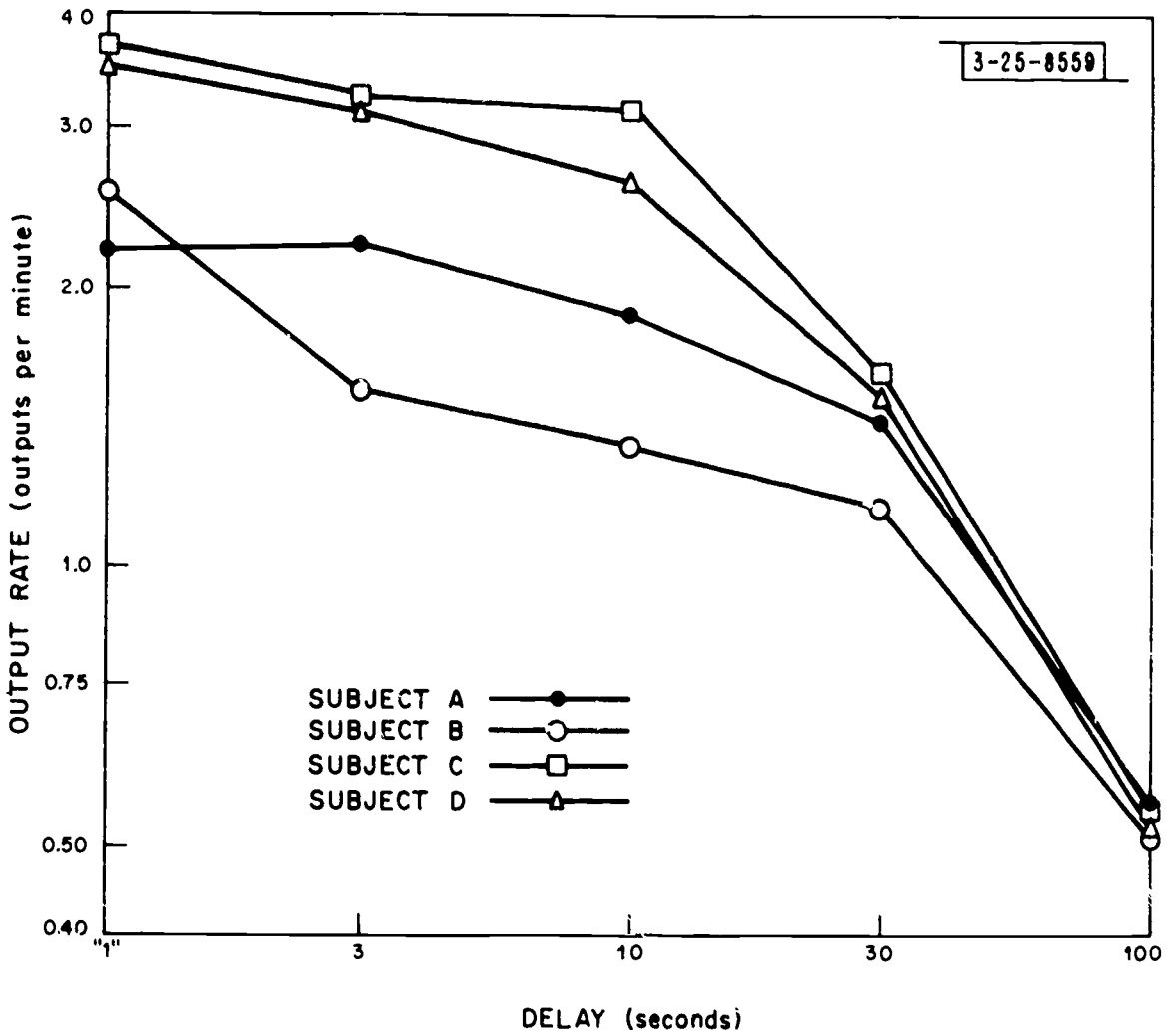


Fig. 3. Load the user puts on the machine: Ratio of mean number of outputs to mean completion time in the RR experiment. (Log scales on both axes.)

TABLE I

Analysis of Variance of Logarithm of Net Completion Time in the RR Experiment

Source	SS	df	MS	Test against	F	Sig. level
Subjects	3.026	3	1.009	Error	28.9	0.001
Delays	3.744	4	0.9360	D × S	16.8	0.001
D × S	0.6694	12	0.05578	Error	1.60	—
Error	2.793	80	0.03491			

TABLE II

Analysis of Variance of Logarithm of Number of Outputs in the RR Experiment

Source	SS	df	MS	Test against	F	Sig. level
Subjects	1.022	3	0.3407	Error	18.6	0.001
Delays	0.2471	4	0.06178	D × S	2.90	0.1
D × S	0.2545	12	0.02121	Error	1.16	
Error	1.464	80	0.01830			

The geometric mean of net completion time for each subject in each delay condition is shown in Fig. 4. The time decreases with longer delays, and as the analysis of variance shows, this effect is reliable. The trend agrees with the subjects' reports of the way they performed the task; i.e., during long delays they were not just waiting idly, but were able to use the time to plan and often to type the next command.

Number of outputs - Table II presents an analysis of variance of the logarithm of the number of outputs. Subjects were treated again as a random factor, and delay conditions as a fixed factor. The table shows that the main effect of subjects was significant, but that the effects of delays and the subject-by-delay interaction were not significant.[7]

The geometric mean of the number of outputs received during the task is shown in Fig. 5, plotted on a logarithmic scale for each subject in each delay condition. As might be expected, the number of outputs declines as the delay increases. According to the analysis of variance, this effect is not quite statistically reliable, but the analysis of variance does not take account of the fact that the delay conditions are ordered. When that fact is considered, it seems that the number of outputs decreased slightly as the delay increased. In other words, the subject apparently

7. An examination of practice effects shows that the residual fell 0.1007 log units in the 25 trials, thus accounting for 6.3% of the error sum of squares. Again an analysis of covariance does not seem worthwhile.

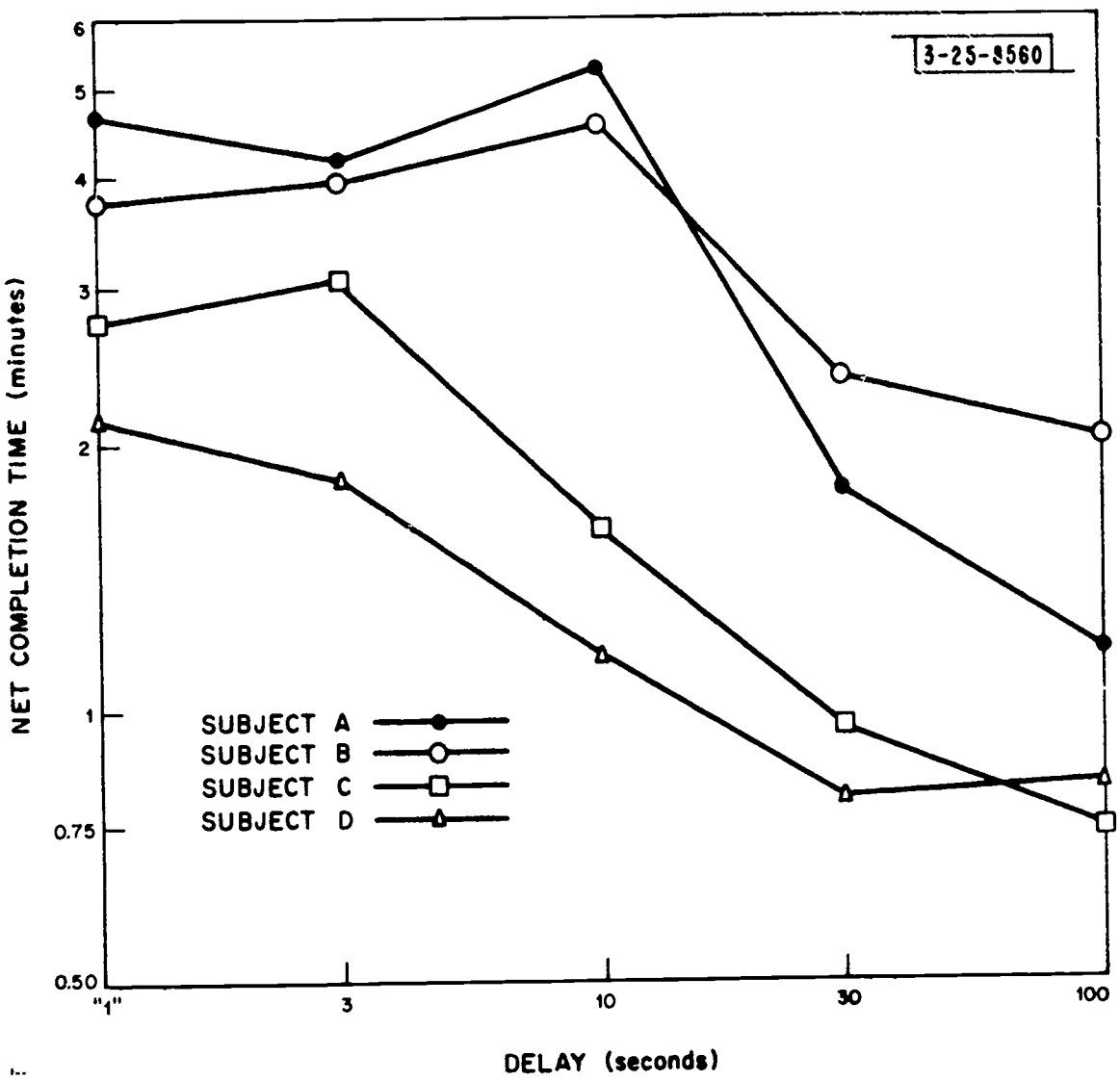


Fig. 4. Geometric mean of net completion time in the RR experiment. (Log scales on both axes.)

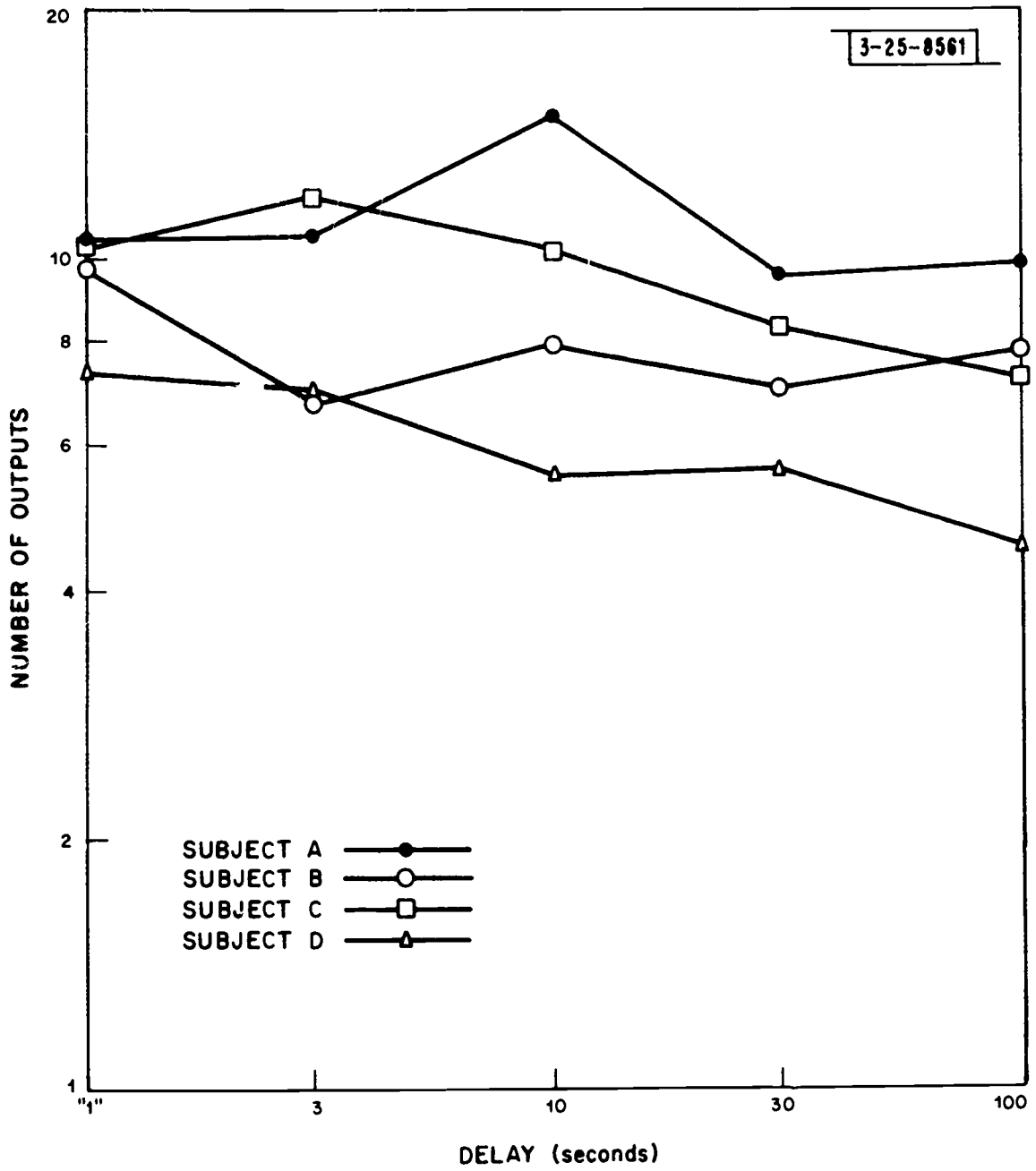


Fig. 5. Geometric mean number of outputs in the RR experiment. (Log scales on both axes.)

managed to use fewer commands when each command incurred a long delay.

EXPERIMENT II: Black Box Tasks

Procedure

The Black Box (BB) tasks were somewhat more complex than the RR tasks, and were designed to have greater face validity: they seem closer to what scientists and engineers actually do on computers. They required more tools of the kind available to the user of the Reckoner facility; instead of the single "bump" command that was needed in the RR task, about a dozen commands were available and were used on the BB task. The BB tasks are also rather similar to some classical tasks used in psychological studies of problem-solving.

When the subject was ready to work on a BB problem he typed a problem number and the computer responded with a time message and a CRT display like that shown in Fig. 6. Three signals are shown in Fig. 6, two inputs (the straight lines) to a simple network and one output (the S-shaped line) from that network. The horizontal axis is a time-axis. The subject's task was to determine the nature of the network, that is, knowing the configuration of the network (which was the same for all problems and is shown in the inset of Fig. 7) to find what specific mathematical operations were required to produce the given output from the two inputs. Any of the eight possible transforms (4 of which are shown in Fig. 7, the other 4 being the inverses of the 4 shown) might occur in either of the two transform boxes of the network, and either of the two possible combining operations (either multiplying or averaging) in the combining circle of the network.

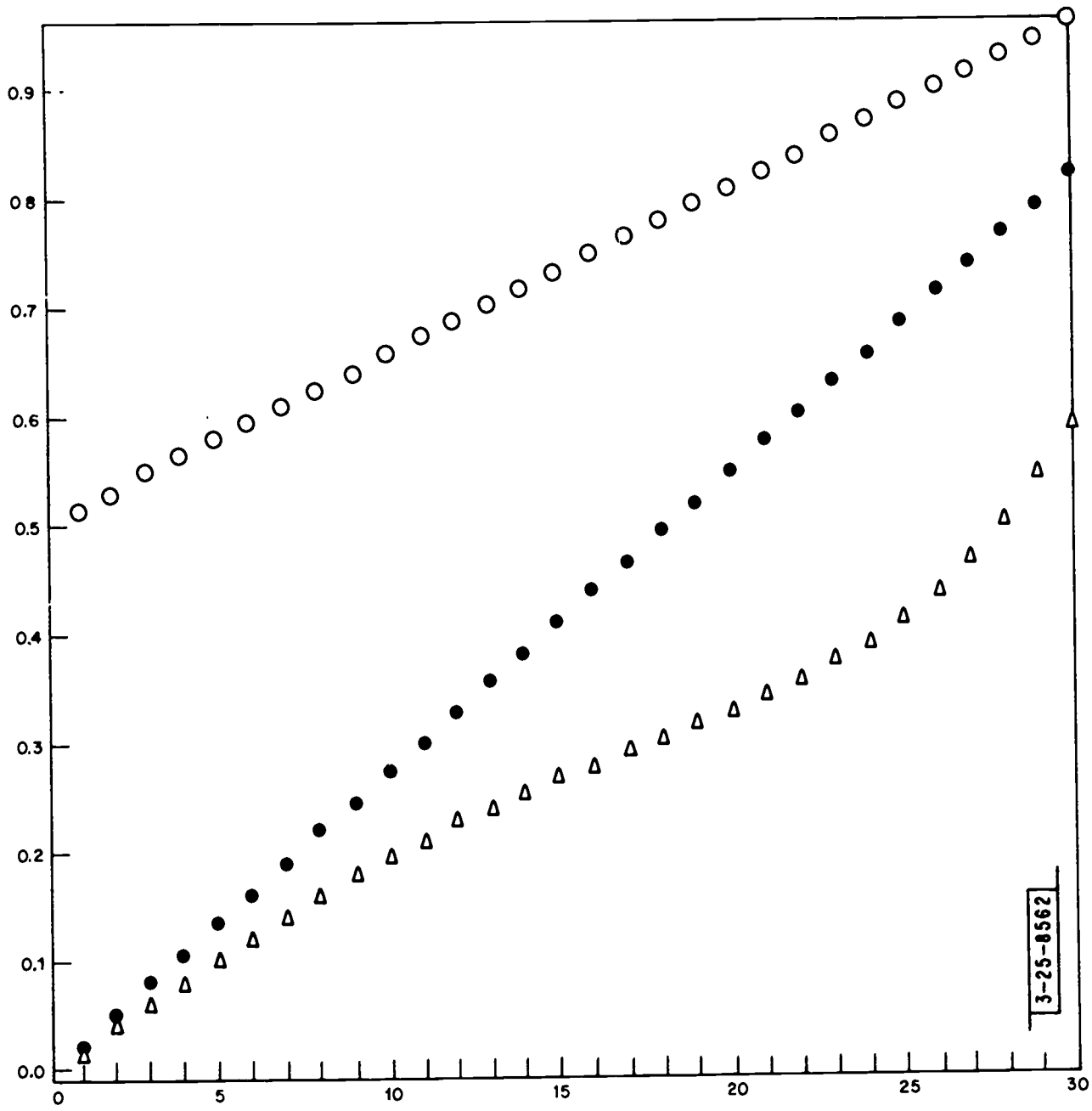


Fig. 6. An example of the initial display in the BB task.

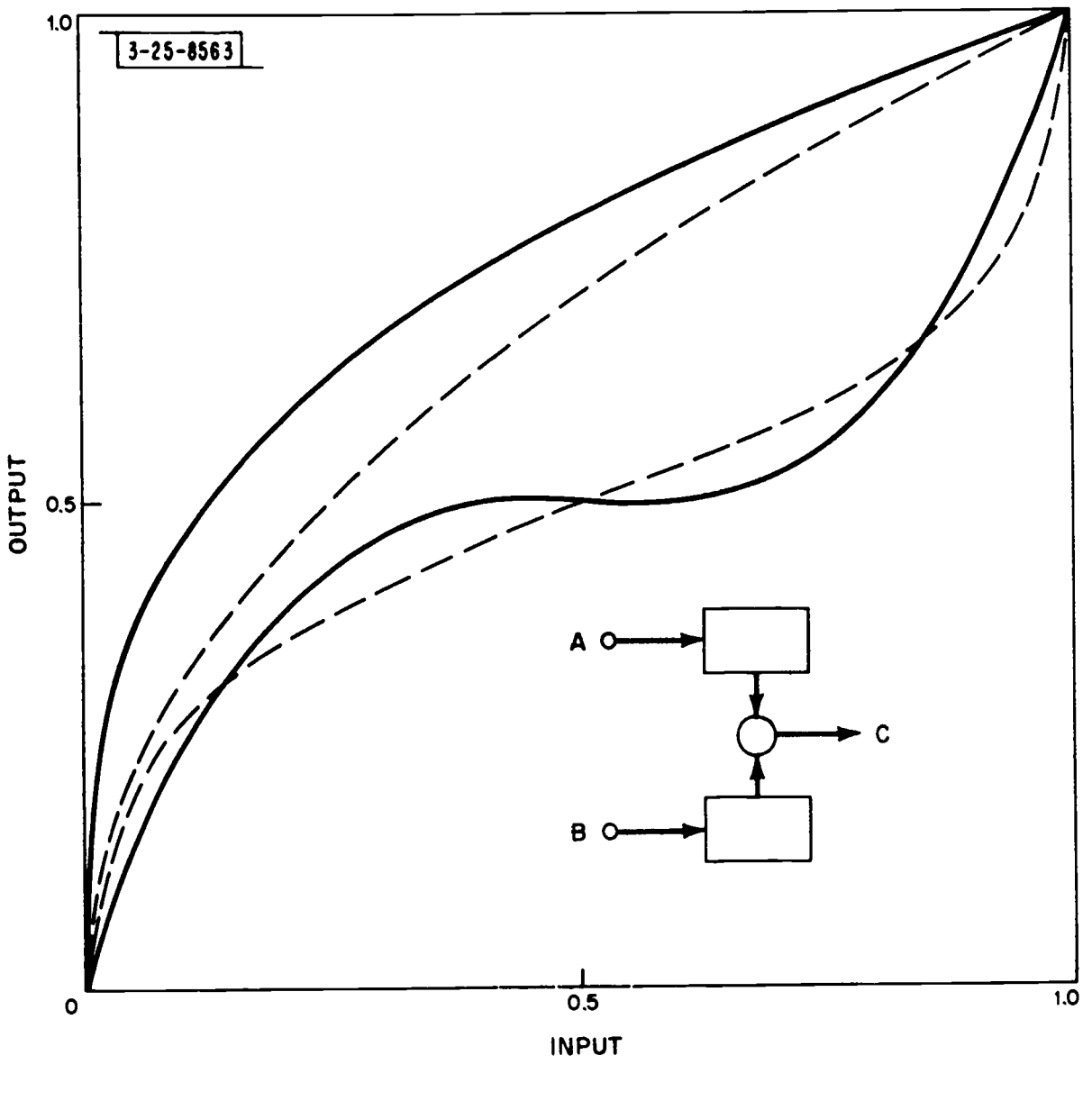


Fig. 7. A graph of four of the transforms used in the BB task, and a diagram of the network.

After preliminary practice and discussion among the subjects, the following procedure evolved as the standard way of attacking the problem. First, the subject guessed at one of the transforms from inspection of the initial display and applied that transform on the appropriate input. Second, the subject, guessing at the combining operator, undid the combining operator by computing the inverse operator (unaverage or divide) on the output and the just transformed input, and displayed the result. If the display seemed similar to one of the possible transforms (which the subject had graphed on a sheet of paper beside him) the problem was essentially solved. The subject computed the appropriate transforms of the inputs, combined them into an output, and tested his result with a special routine called TEST that matched his output with the problem output. If the result was incorrect, the subject received a message telling him so; if the result was correct the time was typed by the computer and the trial ended. (Further details of the BB task are presented in Appendix B.)

Every problem was randomly constructed: in particular, the transforms and the operator specifying a problem were chosen at random as well as the two inputs (See Appendix B.). Delay conditions (1, 3, 10, 30, and 100 sec.) were randomized over sets of 25 problems for each subject, under the restriction that in each set of 25, each subject solved five problems at each delay condition. There were two sets of 25 problems for each subject.

Results

The BB experiment will be analyzed as a 4 by 5 factorial design (4 subjects by 5 delays) with 10 trials per cell – 200 trials in all. Completion time was the time between the appearance of the display that started the task, and the appearance of a time typed out by the TEST process. (See Appendix B.) In this experiment there were a few cases in which a command that did not request a display produced an error-message; so the number of outputs was only approximately equal to the number of displays requested. As before, the display that began the task was not counted in the number of outputs, but the output from the TEST command that ended the task was included.

Completion time and output rate. - The arithmetic mean of the times needed to complete the 10 tasks at each delay condition are shown in Fig. 8 for each subject.* The number of outputs per minute as a function of delay condition are shown in Fig. 9 for each subject.† Again we conclude that an experiment large enough to produce curves smooth enough for a system designer would be quite feasible.

Net completion time. - Table III shows an analysis of variance of the logarithm of the net completion time.‡,[8] Subjects were treated as a random factor and delays

---

* See footnote 3, page 11.

† See footnote 4, page 13.

‡ See footnote 5, page 13.

8. An examination of practice effects shows that the residual fell 0.1459 log units in

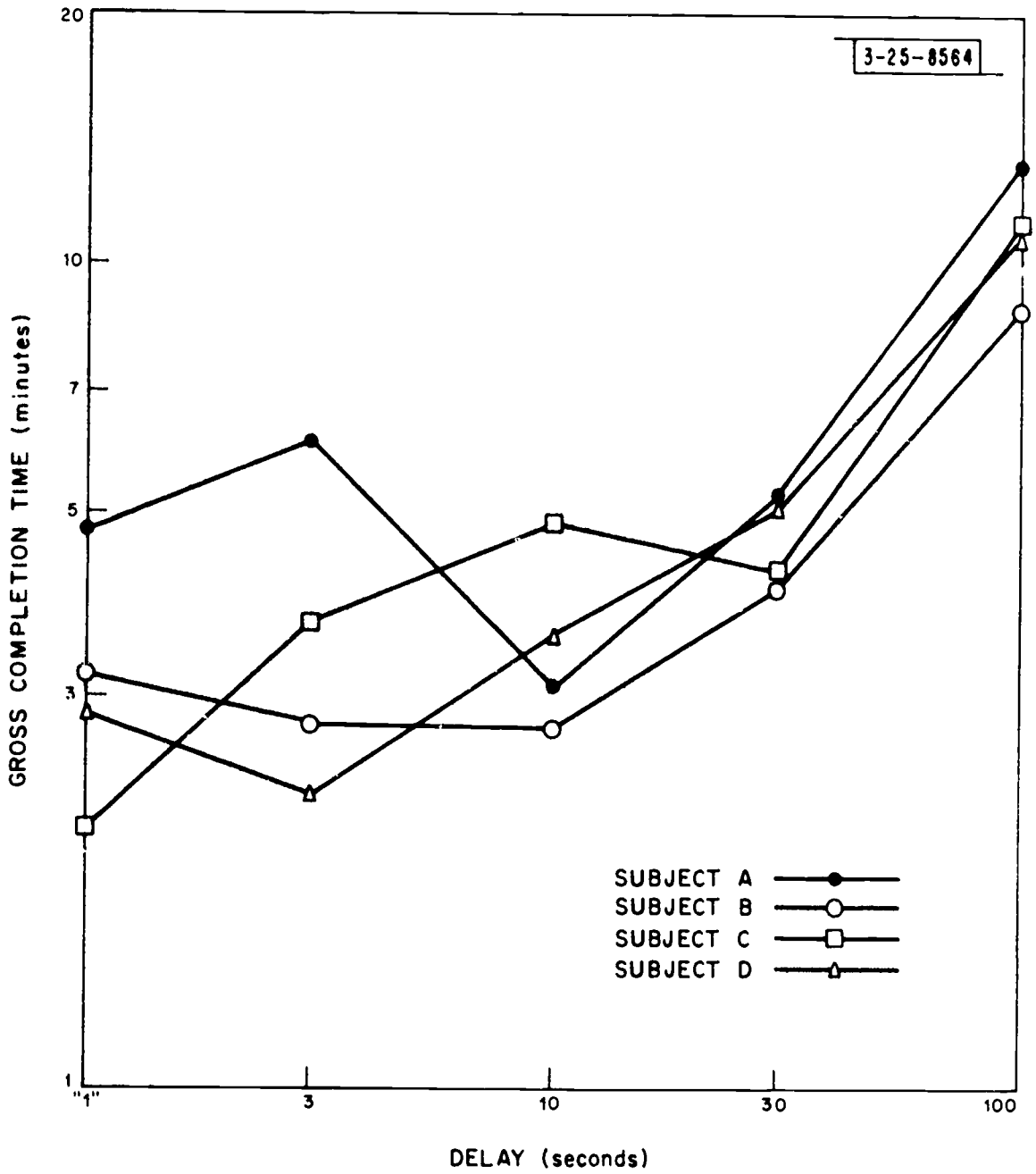


Fig. 8. Expenditure of the user's time: Arithmetic mean of time to complete a task in the BB experiment. (Log scales on both axes.)

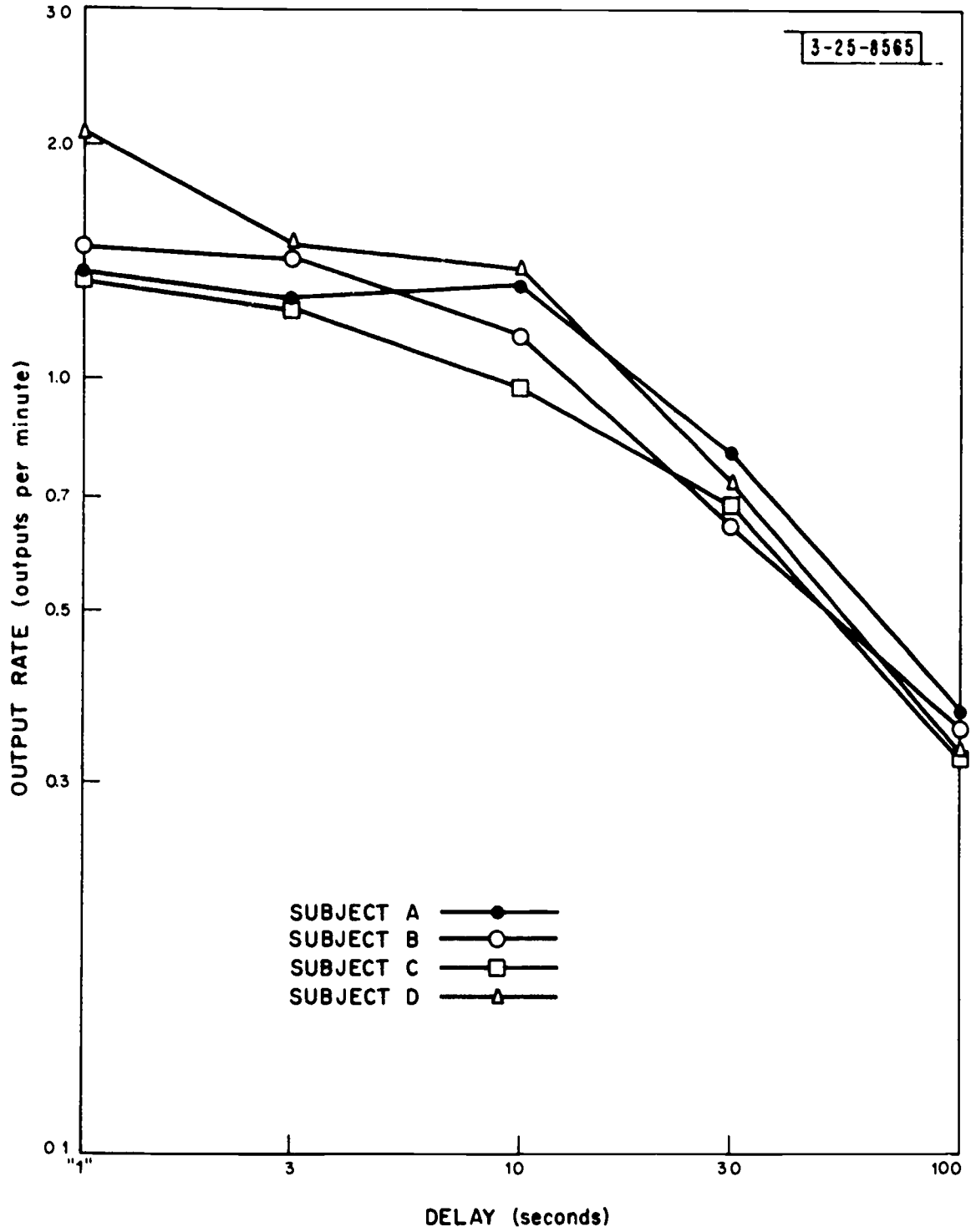


Fig. 9. Approximate load the user puts on the machine: Ratio of mean number of outputs to mean completion time in the BB experiment. (Log scales on both axes.)

TABLE III

Analysis of Variance of Logarithm of Net Completion Time in the BB Experiment

Source	SS	df	MS	Test against	F	Sig. level
Subjects	0.2210	3	0.07367	Error	0.604	—
Delays	1.394	4	0.3484	D × S	5.37	0.025
D × S	0.7790	12	0.06492	Error	0.532*	—
Error	21.98	180	0.1221			

*The significance level of the F of 0.532 is less than 0.9; i.e., it is not significant at the 0.1 level the "wrong" way.

TABLE IV

Analysis of Variance of Logarithm of Number of Outputs in the BB Experiment

Source	SS	df	MS	Test against	F	Sig. level
Subjects	0.5028	3	0.1676	Error	5.36	0.025
Delays	0.3447	4	0.08618	Error	2.75	0.1
Error	0.3756	12	0.03130			

as a fixed factor. The main effect of subjects was not significant, the interaction of subject and delay was not significant, but the main effect of delay condition was significant at the .025 level, even though the analysis of variance did not take account of the fact that the delay conditions are ordered. If that fact is taken into account, the effect of delay is highly significant. In particular, the linear component of the relation between logarithm of nominal delay and main effect of delay is significant at the .005 level when tested against the D × S interaction ($F \approx 13$, with 1 and 12 d. f.).

Figure 10 shows the geometric mean of net completion time for each subject in each delay condition. In this figure, as contrasted with Fig. 4, the net time increases with longer delays. Thus a long delay does more than just make the subject wait; it further degrades his performance in some way.

Number of outputs - Table IV shows an analysis of variance of the logarithm of the number of outputs. * The deviations of the observations from the cell mean (i. e. , from the mean for each combination of subject and a delay) are so badly skewed that they could not be used to compute an error sum of squares. The skewness comes in part from the great number of solutions to the BB task that took place in two output steps. Since the cell means are averages of 10 observations their

---

50 trials, thus accounting for 2% of the error sum of squares. As before, an analysis of covariance does not seem worthwhile.

* See footnote 5, page 13.

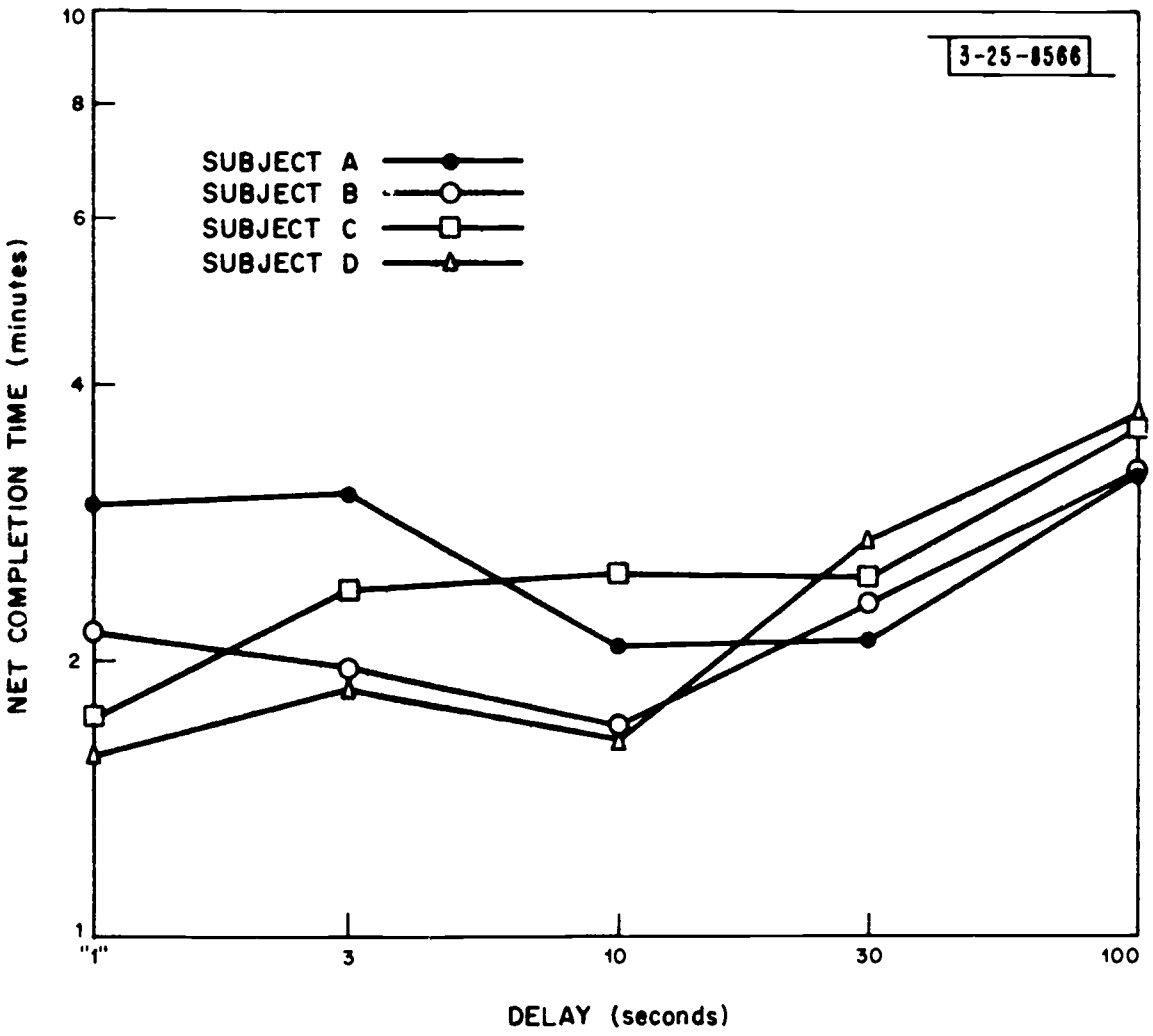


Fig. 10. Geometric mean of net completion time in the BB experiment. (Log scales on both axes.)

distribution should be reasonably normal; therefore the cell mean was regarded as the basic datum for this analysis, and the experiment was treated as a 4-by-5 design with one observation per cell.[9] The effect of subjects is significant, but the effect of delays does not quite reach significance. Even when we take account of the fact that delay conditions are ordered, it is not clear that delay has any effect.

For each subject in each delay condition, the geometric mean of the number of outputs received during the task is shown in Fig. 11. As has just been noted, the number is not much affected by the delay.

---

9. Practice effects were trivial in this case. The within-cell residuals rose 0.1068 log units over the 50 trials, thus accounting for 0.014% of the sum of squares of within-cell residuals.

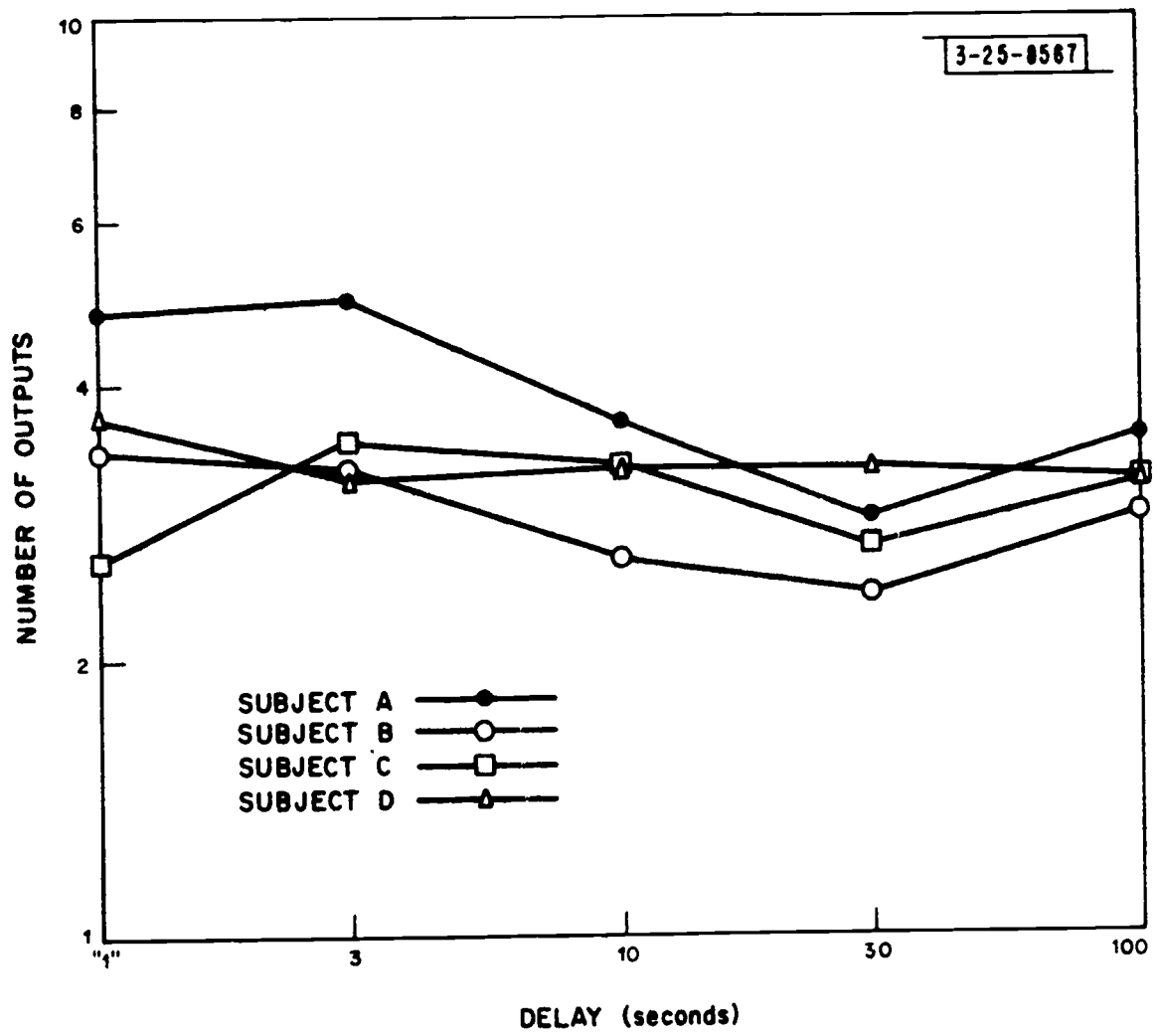


Fig. 11. Geometric mean number of outputs in the BB experiment. (Log scales on both axes.)

EXPERIMENT III: Scattershot Tasks

Procedure

The Scattershot Experiment (SS) was designed to come closer to the kinds of real problem-solving that scientists and engineers would be doing when they used a facility like the Lincoln Reckoner. An attempt was made to select problems that would be unique, diverse, and real—thus the label "scattershots." The subjects had the full resources of the Reckoner, with some minor modifications necessary for controlling the delay in the machine's response, and we tried to select problems of the kind that might arise as sub-tasks in the process of solving real problems on the Reckoner.

The criteria used in selecting the problems were: (1) that the problems should take about ten minutes when the responses of the machine were not delayed, (2) that the problems should not demand specialized training in any particular branch of science or engineering, but should demand the kind of thinking in which a scientist or engineer is engaged when he does computation, and (3) that each problem should have a definite stopping point, so that the quality of the solution need not be considered, only the time required to reach the solution. (Appendix C gives four of the problems that were used in this experiment.)

The subject worked from a loose-leaf notebook that had two pages for each task. The first page usually gave instructions about loading some procedures or files of data that would be used in the problem. When the subject had finished whatever preparations the first page described, and was actually ready to begin work on the

problem, he typed the number of the task. The machine replied by typing out the time at which he had done so, and he turned to the second page, on which he found the statement of the problem he was to solve. He was then free to work on the problem, using the Reckoner in whatever way he wished, subject to whatever restrictions were specified in the statement of the problem. As soon as the subject had produced a correct result, he was required to type the word "DONE," and the computer responded with the time at which he had typed "DONE."

The delay conditions were 1, 10, 30, and 100 sec.; except for the omission of the 3 sec. condition, they were the same as in the previous experiments. There were 16 tasks, four devised by each of the four subjects (who were, as usual, the four experimenters). Each task was performed by all four subjects, each subject working under a different condition of delay; but the data from trials on which a subject was performing his own task were discarded. The data from two other trials were discarded because the subjects had misread the instructions.[10] Thus 46 trials remained to be analysed.

Note that because a task could be used in only three trials (once by each of the subjects who had not devised it), it was not possible to balance the effects of tasks

---

10. In one case the subject failed to finish the task; in the other, the subject used an illegitimate shortcut. Happily, these misfortunes involved different subjects performing different tasks under different conditions.

across the four subjects or across the four delays. This presented a problem because it was expected that the tasks would vary greatly in difficulty. It, therefore, had to be assumed that just taking an average over the trials made by a given subject or under a given condition of delay would not be very meaningful. The average would depend heavily on what tasks were performed on those trials; so it could not meaningfully be compared to averages for other subjects or other delays.

Therefore, the experiment was designed so that it would calibrate the tasks, as well as yielding information about the effects of interest – the effects of delays, subjects and their interaction. Then the calibrations could be used to correct the results so that data from trials on w ich different tasks were performed would be comparable. Now in theory the experiment might have been designed so that some of the trials would be used to calibrate the tasks and the rest would be used to measure the effects of interest. In practice it is more efficient to use each trial partly for the one purpose and partly for the other – in other words, design the experiment so that the two kinds of information can be untangled by solving a set of simultaneous equations. It was for the sake of accuracy in the untangling that the experiment was designed so that each task would be performed under three different delays.

One advantage of using each trial for both purposes is that the experiment probably will not be ruined if a few trials are lost. As we have said, two trials were in fact lost.

Results

The calibration of the tasks was done afresh for each measure of the subject's performance – gross completion time, number of outputs, and net completion time. A multiple linear regression analysis of the logarithm of the measure in question produced a calibration factor for each task (for the details, see Appendix D);* the result of each trial was then adjusted by multiplying it by the calibration factor for the task performed on that trial. The adjusted results were used in plotting the graphs that will be shown here.

The multiple linear regression analyses were also used to test the significance of the effects of delays, subjects, and their interactions. Because of the imbalance in the design, analyses of variance cannot be used to make these tests.

In this experiment the gross completion time is the time between the command "TASK n" that starts the trial and the command "DONE" that ends it. The number of outputs is, as usual, the number of delays the subject suffers in the course of the trial; i. e., the number of commands between "TASK n" and "DONE" that request displays or cause error-messages.

Gross completion time and output rate. - Calibration factors for the tasks

---

* See footnote 5, page 13.

were derived (from the regression analysis explained in Appendix D), and an adjusted completion time for each trial was computed by multiplying the actual completion time by the calibration factor for the task being performed. The arithmetic means of the adjusted completion times for the trials performed by each subject in each condition of delay are shown in Fig. 12.*

Although the tasks varied greatly in difficulty (the calibration factors varied over a range of more than ten to one) the use of multiplicative corrections (i. e., calibration factors) to compensate for differences between tasks appears to have been remarkably successful. The 16 points in Fig. 12 are based on a total of only 46 observations, and about a third of the data (15 of the 46 degrees of freedom) were used up in deriving the calibrations. The fact that the curves in Fig. 12 turn out to be as regular as they are with so few observations per point argues that finding a correction factor for each task is a satisfactory way to keep the experiment under control even when the tasks are very different from each other. To say almost the same thing in another way, in the linear regression analysis, the effects of tasks, subjects, and delays accounted for 91% of the variance of the logarithm of gross completion time.

To show the average output rate in a manner similar to previous figures

---

*See footnote 3, page 11.

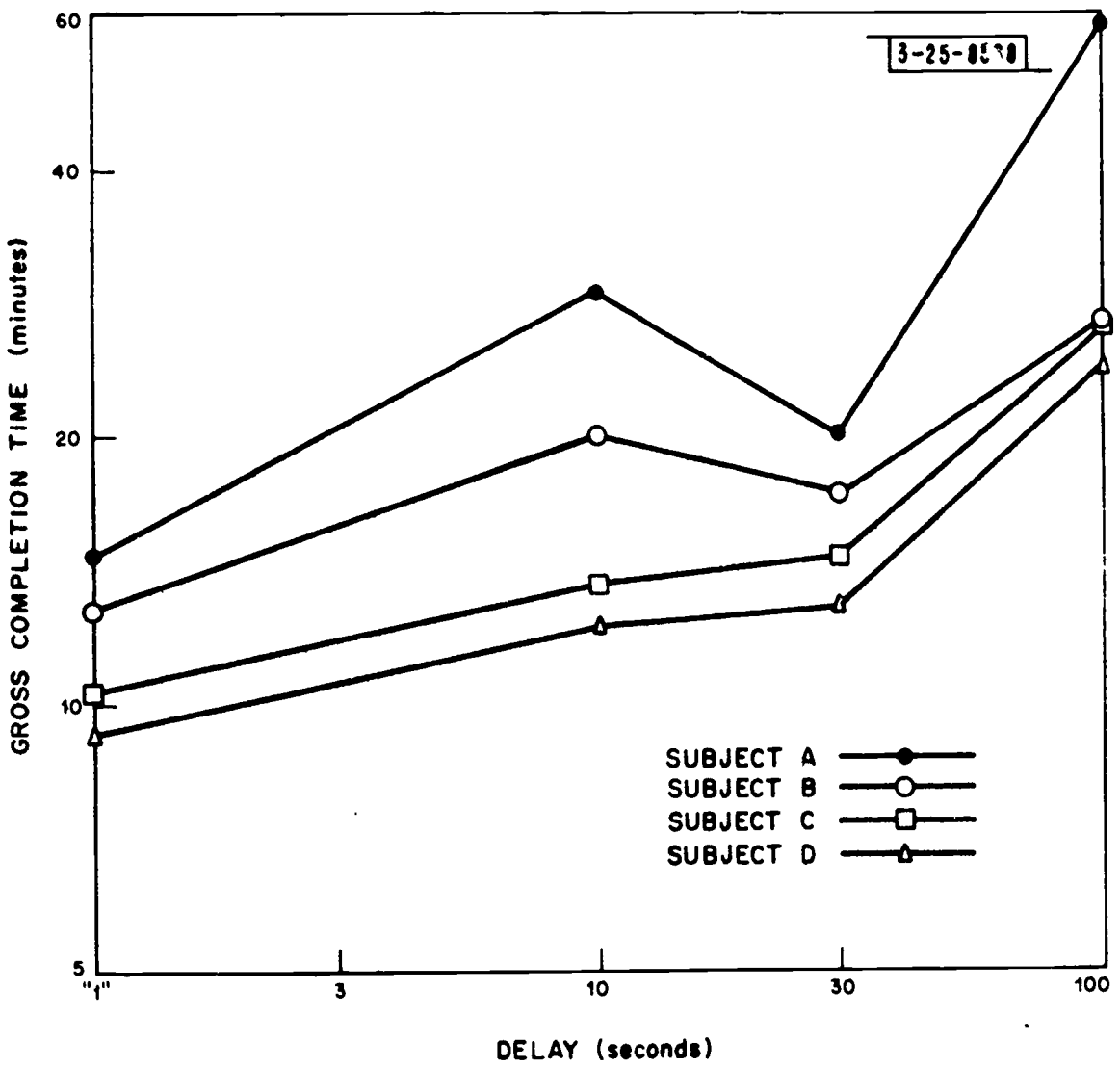


Fig. 12. Expenditure of the user's time: Arithmetic mean of time to complete a task in the SS experiment. (Log scales on both axes.)

(Figs. 3 and 9), the arithmetic mean of adjusted number of outputs (as defined in Appendix D) was computed for each subject in each condition of delay. That mean was then divided by the mean completion time plotted in Fig. 12, and the quotient is the average output rate shown in Fig. 13.

Net completion time - In Table V are the results of the linear regression analysis of the logarithm of net completion time.* Comparison of the various models (as explained in Appendix D) shows no significant delay-by-subject interaction, but highly significant effects of delays, subjects, and tasks.[11,12] Comparison of the residual sums of squares for Models 1, 2, and 4 again suggests that a multiplicative correction for differences between tasks works fairly well. The subject effects (in the order in which their curves appeared in Fig. 12 from top to bottom) are: 0.203, 0.001, -0.063, and -0.141. Thus, taking the antilogarithms of these

---

* See footnote 5, page 13.

11. A problem that does not arise in analyzing the results of this experiment is allowing for the effects of practice. Since all the subjects performed the sixteen tasks in the same order, the effects of practice will just appear as part of the difference between tasks.

12. To be conservative, we should point out that if the tests had been done in a different order (if, for instance, Model 3 in Table VI had been used in place of the present Model 3), the effects of delays would have been significant at only the 0.1 level. However, we conclude that the effect of delay is real. If we compare to Model 2 the following model, $\mu + \alpha_T + \delta \log \tau + \epsilon$, where τ is the nominal delay, we find that δ is significantly different from zero at the 0.02 level ($F = 6.40$, with 1 and 29 d.f.).

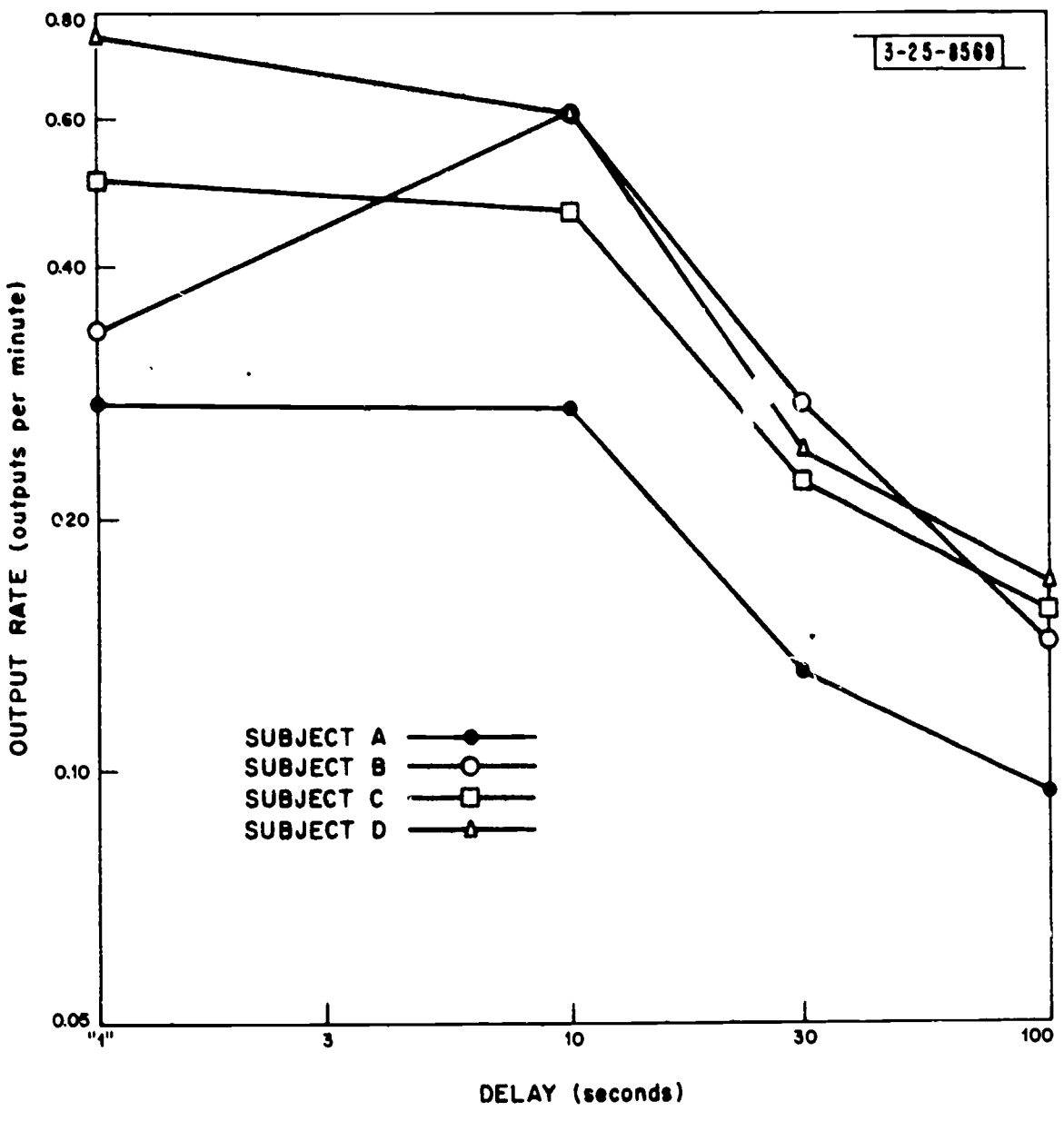


Fig. 13. A rough approximation to the load the user puts on the machine: Ratio of mean number of outputs to mean completion time in the SS experiment. (Log scales on both axes.)

TABLE V

Multiple Linear Regression Analysis of Logarithm of Net Completion Time in the SS Experiment

Model	Residual SS and its df		F and its df		Sig. level	Factor tested
(1) $\mu + \epsilon$	5.628	45	5.22	15 & 30	0.0001	Tasks
(2) $\mu + \alpha_T + \epsilon$	1.559	30	5.80	3 & 27	0.005	Subjects
(3) $\mu + \alpha_T + \gamma_S + \epsilon$	0.9479	27	5.01	3 & 24	0.01	Delays
(4) $\mu + \alpha_T + \beta_D + \gamma_S + \epsilon$	0.5830	24	0.713	9 & 15	—	S × D
(5) $\mu + \alpha_T + \beta_D + \gamma_S + \lambda_{DS} + \epsilon$	0.4082	15				

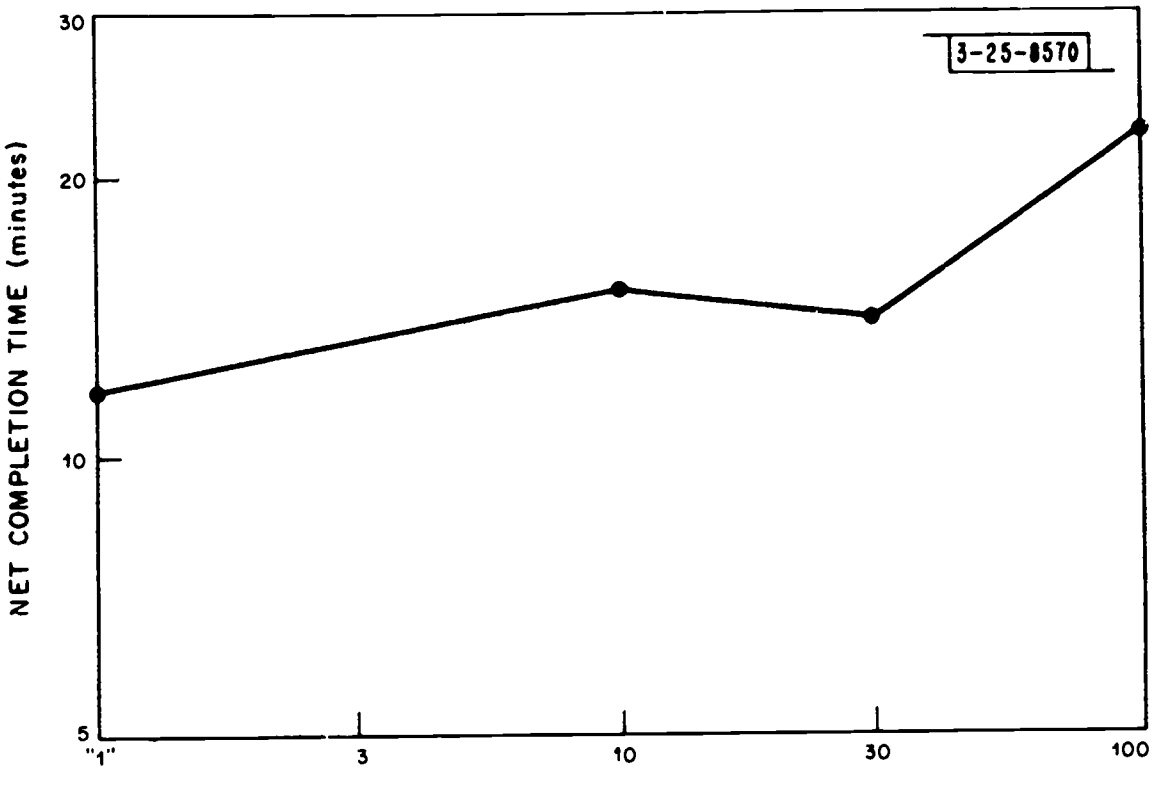
Note: In each F-test, the model in that line is used as the null hypothesis, and the model in the next line as the alternative hypothesis.

numbers, the fastest subject was 2.2 times as fast as the slowest.

In Fig. 14 are presented the geometric means of the adjusted net completion times for each condition of delay, weighting subjects rather than trials equally because there were significant differences between subjects. The adjusted net completion times were obtained (as explained in Appendix D) by deriving calibration factors and multiplying the calibration factor for the task being performed by the net completion time on each trial. In Fig. 14 there appears a significant increase in net completion time as a function of delay, a result similar to that obtained in the Black Box experiment (Fig. 10), but different from that obtained in the Railroad Track experiment (Fig. 4).

Number of outputs. - In Table VI are presented the results of the linear regression analysis of the logarithm of the number of outputs. Comparison of the appropriate models (as explained in Appendix D), reveals that the delay-by-subject interaction and the differences between subjects are not significant, but that the differences between delay conditions and the differences between tasks are highly significant. As before, a multiplicative correction for differences between tasks seems to work well: compare the residual sums of squares for Models 1, 2, and 3.

In Fig. 15 the geometric means of the number of outputs are presented for each delay condition. The linear regression analysis (see Appendix D) yielded a calibration factor for each task, and the number of outputs observed on a trial was



DELAY (seconds)

Fig. 14. Geometric mean of net completion time, averaging over subjects, in the SS experiment. (Log scales on both axes.)

TABLE VI

Multiple Linear Regression Analysis of Logarithm of Number of Outputs on a Trial in the SS Experim

Model	Residual SS and its df		F and its df		Sig. level	Factor tested
(1) $\mu + \epsilon$	7.856	45	6.23	15 & 30	0.0001	Tasks
(2) $\mu + \alpha_T + \epsilon$	1.908	30	6.15	3 & 27	0.005	Delays
(3) $\mu + \alpha_T + \beta_D + \epsilon$	1.133	27	0.223	3 & 24	—	Subjects
(4) $\mu + \alpha_T + \beta_D + \gamma_S + \epsilon$	1.103	24	0.779	9 & 15	—	D × S
(5) $\mu + \alpha_T + \beta_D + \gamma_S + \lambda_{DS} + \epsilon$	0.7515	15				

Notes: In each F-test, the model in that line is used as the null hypothesis, and the model in the next line as the alternative hypothesis. The significance level of the F of 0.223 is less than 0.9; i.e., it is not significant at the 0.1 level the "wrong" way.

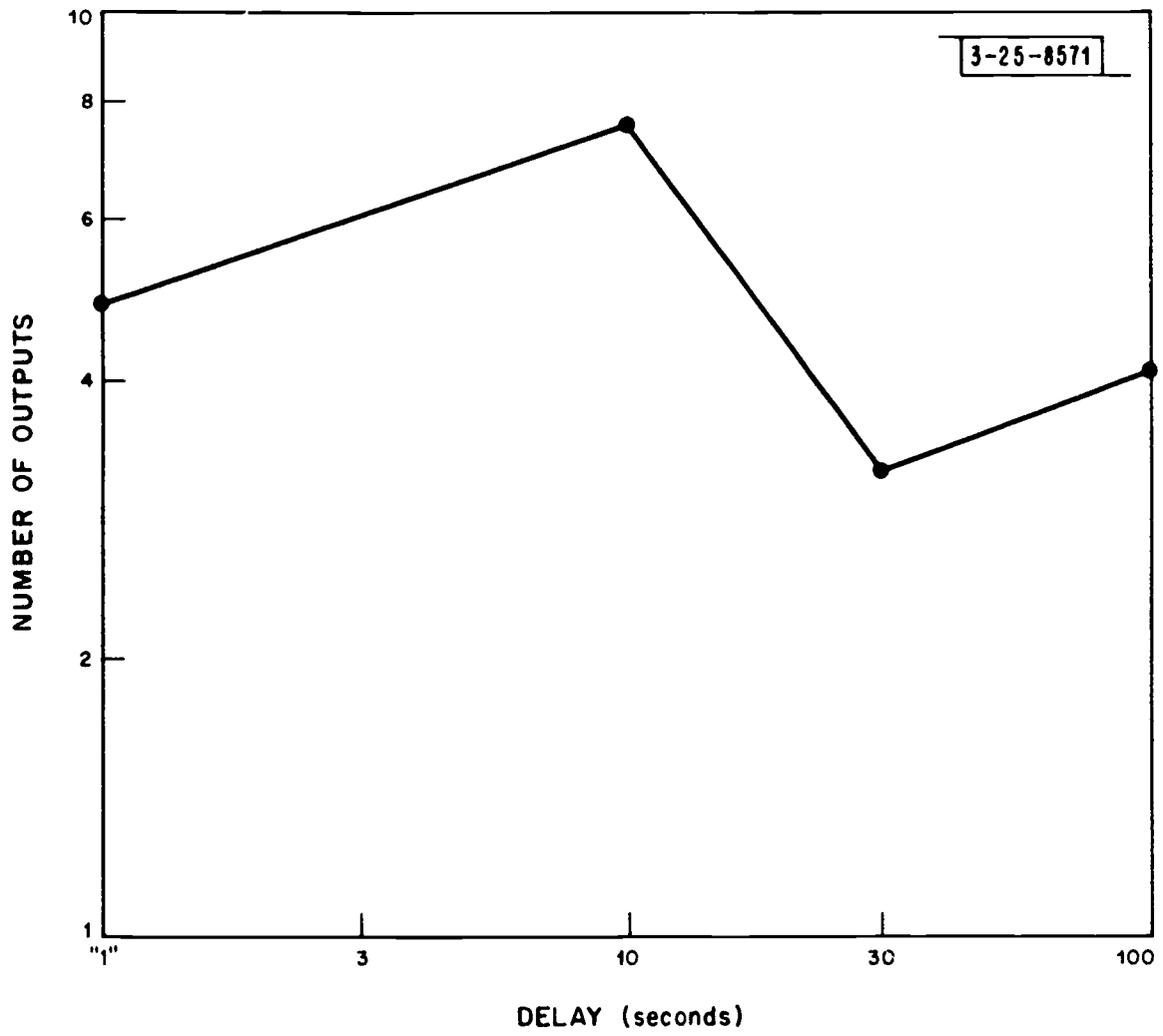


Fig. 15. Geometric mean number of outputs, averaging over subjects, in the SS experiment. (Log scales on both axes.)

multiplied by the calibration factor for the task the subject was performing, thus producing the adjusted number of outputs. The geometric mean of the adjusted number of outputs was computed for all trials made under a delay condition, weighting all the trials equally since there were no significant subject differences.

The form of the curve shown in Fig. 15 is baffling. It would be reasonable to suppose that the number of outputs would decline as the delay increases (the subject would try to get along with a smaller number of outputs when outputs are more expensive), but it is difficult to understand why the maximum number of outputs should occur when the delay is 10 sec. We doubt that the maximum at 10 sec. is real, but because the effect of delay is statistically significant, we feel obliged to watch for similar results in future experiments.

DISCUSSION

The primary importance of the preceding three experiments is that they indeed demonstrate the feasibility of obtaining functional quantitative relations in human factors experiments on man-computer interaction, specifically, in experiments on the delay in the machine's response.

Furthermore, the gross completion time curves from the various experiments (Figs. 2, 8, and 12) give some preliminary indication of the way the amount of time the user must expend depends on the delay in response.* The output rate curves (Figs. 3, 9, and 13) give some indication of how the delay affects the load the user puts on the system, although it should be remembered that in the third experiment number of commands rather than number of outputs might have been a more appropriate measure of the actual computation load. Although the present curves may not be smooth enough so that design decisions could be based on them, they imply that experiments large enough to produce stable curves would be quite feasible.

From the point of view of experimental technique, one of the most important findings is that with realistic tasks of the kind used in the Scattershot experiment, multiplicative corrections – i. e., calibration factors – are a good way to compensate for differences between tasks. This is a useful finding because realistic tasks will inevitably vary considerably, and without some way of compensating for the variations, realistic experiments would be much more difficult to conduct.

It was not a foregone conclusion that multiplicative corrections for differences

---

*But remember that the Reckoner is unusual: it "stacks" commands. (cf. p5)

between tasks would be useful. It seems reasonable to suppose that if Task A takes a slow subject twice as much time as Task B does, then Task A will take a fast subject twice as much time as Task B does. But is it reasonable to suppose that if Task A takes twice as much time as Task B with short delays, it will take twice as much time with long delays? Probably not. But to be useful, a correction for differences between tasks does not have to be exactly right; it only has to be approximately right.

Differences between the subject's behavior in the various tasks are seen more clearly in such indices of performance as net completion time and number of outputs. Net completion time shows rather directly the effects of delay on performance: if net completion time increases the subject is being "distracted" by the delay, whereas, if net completion time decreases the subject is making use of the delays. Both effects occurred in the present three experiments: in the RR tasks net completion time decreased as a function of delay, whereas in the BB and SS tasks this variable increased. Since the latter two tasks are presumably closer in content to the real tasks users do on on-line systems, it is probably safer to conclude that the effects of delay are generally detrimental to user performance. It could be argued, however, that for some special kinds of on-line computations analogous to the RR tasks (perhaps on-line design work) the segmental nature of the tasks would allow the user to plan a step or two ahead while the machine was preparing an output at longer response-delays.

It may be noticed that in all three experiments the subjects displayed a remarkable relative consistency of performance, independent of task, with respect to each other. Moreover, despite a large range of abilities the overall conclusions seem to apply to all the subjects, thus improving the generality of the findings.

The data on number of outputs generally showed, contrary to what one might expect, that subjects did not very much adjust their number of outputs to conform to delay conditions. That is, the subject does not decrease the number of outputs even when outputs are costly to him. That there exists some tendency to so adjust is hinted at in the results of the RR task, but in the other two tasks the effect is either very small or difficult to interpret, as in the rise in number of outputs at 10 sec. delay in the SS tasks.

Perhaps the techniques of experimentation used in this note can be applied to other questions about computer system design. In particular, such questions as "stacking" of inputs, utility of scope displays, and "compactness" of language may be amenable to such an experimental attack. A question directly related to the present note is the effect of a more realistic, large variance of response-delay on subject performance. The effects may be quite different.

The fact that these experiments have proven to be feasible has implications for new directions in human factors engineering, and opens new kinds of phenomena for investigation, particularly in the area of computing systems (6, 7). It may well be

time for human factors engineers to move beyond knobs, dials, and scope faces and on to cognitive processes. For totally new kinds of problem-solving environments, such as the Reckoner, we are forced to make good guesses as to what will prove useful to people, but it might be more economical to have better conceptions of what people do.

REFERENCES

1. Grant, E. E., & Sackman, H.
   An exploratory investigation of programmer performance under on-line and off-line conditions.
   IEEE Trans. on Human Factors in Electronics, 1967, 8, 33-48.
2. Sackman, H., Erikson, W. J., & Grant, E. E.
   Exploratory experimental studies comparing online and offline programming performance.
   Comm. ACM, 1968, 11, 3-11.
3. Schatzoff, M., Tsao, R., & Wiig, R.
   An experimental comparison of time sharing and batch processing.
   Comm. ACM, 1967, 10, 261-265.
4. Forgie, J. W.
   A time- and memory-sharing executive program for quick-response, on-line applications.
   AFIPS Conf. Proc., 1965, 27 Part 2, 127-139.
5. Stowe, A. N., Wiesen, R. A., Yntema, D. B., & Forgie, J. W.
   The Lincoln Reckoner: an operation-oriented, on-line facility with distributed control.
   AFIPS Conf. Proc., 1966, 29, 433-444.
6. Nickerson, R. S., Elkind, J. I., & Carbonell, J. R.
   Human factors and the design of time sharing computer systems.
   Human Factors, 1968, 10, 127-133.
7. Carbonell, J. R., Elkind, J. I., & Nickerson, R. S.
   On the psychological importance of time in a time sharing system.
   Human Factors, 1968, 10, 135-142.

APPENDIX A

Procedural Details in the Railroad Track Experiment

The Gaussian "bump" routine is specified by the following equation:

$$Y = H e^{-2.0100125(\frac{X-C}{D})^2}$$

in which H is the height of the bump, C is the location along the x-axis, and D is the width, in the sense that 95.5% of the area of the bump lies between C - D and C+D. A sample command by a subject using this routine is:

B .5 -.2 .3

in which the letter B names the routine which calculates the bump, cumulates this bump with previously specified bumps, and displays the result. The parameter .5 specifies the location of the center of the bump (C in the above equation) along the x-axis, which ranged from -1 to +1 . The parameter -.2 specifies the height of the bump (H in the above equation) on the y-axis, which also ranged from -1 to +1 . In this example the height is negative indicating that this bump should be subtracted from the previous position of the line, or any previously specified bumps. The parameter .3 specifies the width of the bump (D in the above equation): 95.5% of the area will lie within a range of .3 on either side of the center-point.

A third degree polynomial determined the locus of the center of the parallel curves. The three coefficients of the polynomial were independently and randomly drawn from rectangular distributions: for the third degree term the range of the

distribution was ±8; for the second degree term, ±4; and for the first degree term, ±2. The two parallel curves were generated by adding and subtracting a small constant (0.01) along the normal to the center curve (which was not displayed). The curves were then scaled so that their x and y extrema were at -1 and +1, giving the same co-ordinate range for all problems.

APPENDIX B

Procedural Details in the Black Box Experiment

Each of the two inputs to the network were linear functions of time, randomly and independently selected. The numbers in each input set were evenly spaced in 30 intervals with a range and minimum value that were randomly selected in the following manner:

R, the range, was randomly chosen from the interval .5 to 1, i.e.,

$$.5 \leq R \leq 1 \ ,$$

and X_{min}, the first value in the set, was then chosen randomly from the interval 0 to 1 - R, i.e.,

$$0 \leq X_{min} < 1 - R \ .$$

The structure of the BB task was further randomized by randomly and independently selecting which transform went with which input set, and which operator combined them.

The commands available to the subject were:

SEE Y X – A command to display, on x - y scales running from 0 to 1, each element of one array of numbers against the corresponding element of another array. Up to three pairs of arrays may be displayed: thus the command could be "SEE Y X N M H G."

AV X Y Z – A command to add each element of the array X to the corresponding element of the array Y, divide each of the sums

by 2, and call the resulting array of numbers Z.

UNAV X Y Z – A command to multiply each element of the array X by 2, add to the product the corresponding element of the array Y, and call the resulting array of numbers Z.

MUL X Y Z – A command to multiply each element of the array X by the corresponding element of Y and call the resulting array of numbers Z.

DIV X Y Z – A command to divide each element of the array X by the corresponding element of the array Y and let the resulting array be called Z.

TEST X Y – A command to compare each element of the array X with the corresponding element of the array Y. If the arrays are the same, a time is typed out; if not, a message "TRY AGAIN" is typed out.

T1 X Y through T8 X Y – Eight commands that perform the transformations shown in Fig. 7 and the inverses of those transformations. Each element of the array X is transformed, and the resulting array is called Y.

Note: In the foregoing, the names "X," "Y," etc., are only examples. In

practice, the subject might use in their place names like "TRYA1B" or "Q2W" – i.e., array names that he made up himself.

APPENDIX C

Examples of Problems Used in the Scattershot Experiment

Task 4: Build a process that will compute N! without any loops.

If the process is named PROC, typing

PROC N F

must create a scalar F that is exactly equal to N! when N is a small, non-negative integer.

Hint: Use the fact that

log (N!) = log 1 + log 2 + log 3 . . . + log N,

and then use γ RND.

Verify that: 0! = 1

2! = 2

5! = 120

9! = 362,880

and then type DONE.

Task 5: The 1 × 500 array, GR5 , contains numbers ranging in value from 0 to 1. Your task is to delete all numbers which are equal to 0.25, in order to form a smaller array, RESULTGR5. Ordinal characteristics of GR 5 are to be retained. *

---

* That is, the rest of the elements of the array were to remain in their original order.

Verification: Type TESTGR5. This process responds with either FIND ERROR or FINISHED. If the latter occurs, proceed quickly to type DONE.

Task 6: Construct an upper triangular matrix of order 20, that is, a 20×20 array in which every element above the diagonal is 1 and every element below the diagonal is 0, and the diagonal itself is 1. When you have verified your answer type DONE.

Task 16: Consider the positive values of x at which

$$10^{-x/20} \sin X = 0.1 \ .$$

At which of those values is

$$f(x) = \frac{(x - 8)^2}{x + 1}$$

smallest? The task is to find f(x), correct to only two significant figures, at that value of x .

Then type DONE.

APPENDIX D

Details of Analysis of Results of the Scattershot Experiment

The linear regression analysis will be explained by reference to Table VI. Model 5, in the last line of the table, assumes that the logarithm of the number of outputs on a given trial is the sum of: a constant μ, a term α_T that depends on the task, a term β_D that depends on the delay condition, a term γ_S that depends on the subject, a term λ_{DS} that depends on both delay and subject, and a random variable ϵ that is normally distributed with expectation zero and variance constant from trial to trial. As usual, the following conventions are adopted:

$$\sum_T \alpha_T = \sum_D \beta_D = \sum_S \gamma_S = 0 ,$$

$$\sum_D \lambda_{DS} = 0 \qquad \text{for any S,}$$

$$\sum_S \lambda_{DS} = 0 \qquad \text{for any D.}$$

The values of μ, the α's, the β's, the γ's, and the λ's, were chosen to minimize the sum of the squares to the 46 residuals. (A residual is the difference between the value of the dependent variable on a given trial and the value $\mu+\alpha_T+\beta_D+\gamma_S+\lambda_{DS}$ assumes when T, D, and S are what they were on that trial.) The minimum of the sum of the squared residuals is shown in the second column of Table VI, and the degrees of freedom of that sum of squares is shown next to it. (There were

46 observations, and we are adjusting 31 parameters when we pick the α's, β's, γ's, λ's, and μ; the difference between 46 and 31 is 15. The degrees of freedom of the residual sum of squares in Model 5 is thus 15.)

Model 4 was examined next; i.e., the least squares procedure was repeated with the constraint that $\lambda_{DS} = 0$ for every D and S. The results are shown in the fourth line of the table, along with the result of an F-test in which Model 4 was treated as the null hypothesis and Model 5 as the alternative hypothesis. As the last column of the table shows, this is a test of the significance of the delay-by-subject interactions, λ_{DS}. The test fails to reject the null hypothesis, so we tentatively accept Model 4 and proceed.

Model 3 was examined next; i.e., the least squares procedure was repeated with the additional constraint that the subject effect, γ_S, be zero for every subject. The F-test in the third line of the table shows that Model 3 does not differ significantly from Model 4: differences between subjects are not statistically significant. In the same fashion, Model 3 was tested against Model 2, and Model 2 against Model 1. In both cases the difference is highly significant: differences between delay conditions are highly significant and so are differences between tasks.

We concluded that Model 3 is the best description of our dependent variable, and so the values of α_T obtained in fitting Model 3 were used as the calibration of the tasks. More precisely, the anti-logarithm of $-\alpha_T$ was taken to be the calibration

factor by which the number of outputs observed in Task T should be multiplied to make the results from various tasks comparable. (The fact that these factors ranged from 0.21 to 5.3 shows how important it was to take account of differences between tasks.)

The analysis of the logarithm of net completion time which is shown in Table V was performed in a similar manner. Comparison of Model 5 with Model 4 shows no significant delay-by-subject interactions; comparison of Model 4 with Model 3, 3 with 2, and 2 with 1, shows highly significant effects of delays, subjects, and tasks. Thus, we conclude that Model 4 is the best representation of the logarithm of net completion time. The geometric means shown in Fig. 14 are, therefore, the anti-logarithms of $\mu + \alpha_T$, where μ and the α's have the values obtained in fitting Model 4.

A similar analysis of the logarithm of gross completion time concluded that Model 4 is the best representation. (This analysis was not presented in a table in the body of the text.) The values of α_T obtained in fitting Model 4 were therefore used to calibrate the tasks on gross completion time, the calibration factor for Task T being the anti-logarithm of $-\alpha_T$.