A COMPUTER SIMULATION EXPERIMENT OF SUPERVISORY CONTROL OF REMOTE MANIPULATION.

SIMON GEORGE McCANDLISH

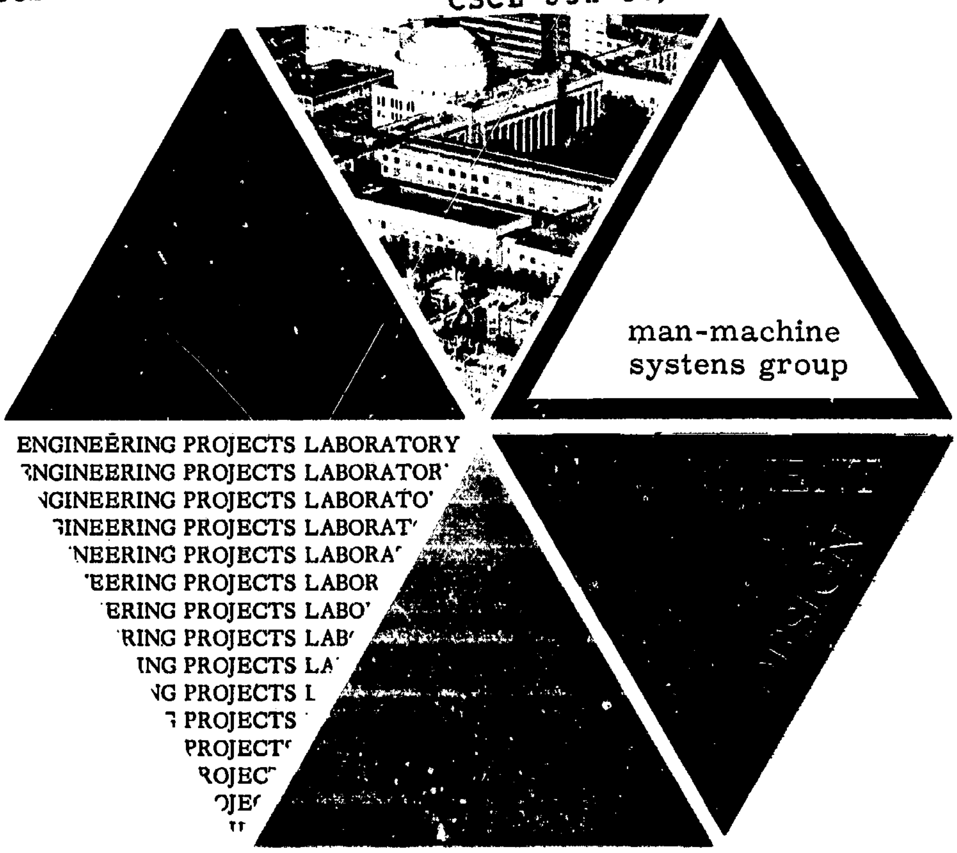
JUNE 1966

Engineering Projects Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology

USAF 19(628)-3317

NASA Grant ~~107-61~~ NGR 22-009-002

DSR 9960-2



A COMPUTER SIMULATION EXPERIMENT OF SUPERVISORY CONTROL OF REMOTE MANIPULATION

by

SIMON GEORGE M[c]CANDLISH

B.A. (Mechanical Sciences Tripos, Pt.I) Cambridge University
1958

M.A., Cambridge University
1962

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Master of Science
in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF
TECHNOLOGY

June 1966

Signature of Author . . . . Simon G. McCandlish . . . . .
Department of Mechanical Engineering, June 1966

Certified by . . . . Thomas B. Sheridan . . . .
Thesis Supervisor

Accepted by . . . .
Chairman, Departmental Committee
on Graduate Students

Before you make of a case a general rule,
test it two or three times and observe
whether all experiments produce identical
results.

Leonardo da Vinci

ABSTRACT

A COMPUTER SIMULATION EXPERIMENT OF SUPERVISORY

CONTROL OF REMOTE MANIPULATION

by

SIMON GEORGE M^{c}CANDLISH

Submitted to the Department of Mechanical Engineering

in partial fulfillment of the requirements for the degree

of Master of Science in Mechanical Engineering.

The replacement of men by remotely operated manipulators is desirable in hazardous task environments such as undersea or interplanetary space. The problems associated with remote operation show that there may be advantages in substituting supervisory control for direct continuous control by the human. This supervisory control requires some low-level intelligence at the remote manipulator.

This report describes a computer simulation of a remote manipulation task and a rate-controlled manipulator; into the latter was built some low-level automatic decision making ability which could be used at the operator's discretion to augment his direct continuous control.

The report describes some experiments on the effect of transmission delay, dynamic lag and intermittent vision on human manipulative ability. The results of these support earlier work which suggested that delay would not make remote manipulation impossible. They also showed that intermittent visual feedback, and the absence of rate information in the display presented to the operator do not seem to impair the operator's performance. These results suggest that a small-capacity visual feedback channel may be sufficient for remote manipulation tasks, or that one channel might be time-shared between several operators.

This report describes further experiments in which the

operator called in sequence various on-site automatic control programs of the machine, and thereby acted as a supervisor. The results suggest that the supervisory mode of operation has some advantages when the task to be performed is difficult for a human controlling directly.

Thesis Supervisor:

Thomas B. Sheridan, Associate Professor of Mechanical Engineering

ACKNOWLEDGEMENT

The author wishes to thank Professor Thomas B. Sheridan, his thesis advisor, who originally suggested this topic. His enthusiasm and understanding have encouraged the author throughout this work.

The author also wishes to thank Professor W. B. Ferrell of the Man-Machine Systems Laboratory for his help and advice.

The author is deeply grateful to three of his friends who gave several hundred hours of their time to act as his subjects. They are Herr Dipl.-Ing. Bernhard F. Fabis of the Institut fur Flugfuhrung at the Technische Hochschule, Braunschweig, and Mr. David J. Barber and Mr. Khushroo M. Captain of the Massachusetts Institute of Technology.

The author wishes to thank the Research Laboratory of Electronics at M.I.T. for the use of their PDP-1 Computer during these experiments.

Miss Jane Benson typed this report from my manuscript for which I am very grateful.

This work was supported in part by the United States Air Force Electronic Systems Division under Contract AF 19(628)-3317, and in part by the National Aeronautics and Space Administration under Grant NsG 107-61.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The desire to be freed of the necessity of working is shared by much of mankind. Those lucky or clever enough to be in positions of power and influence have organized and directed the efforts of those less able than themselves. The twin advance of an organized mercantile system and technical skill introduced the motive and ability to replace human effort by mechanical effort. There are now portions of the world where very little hard physical effort is done by humans. However, those physical tasks requiring nonrepetitive actions and some adaptivity are performed by humans.

Another skill possessed by humans is decision making. The steam engine governor of the eighteenth century is a well known example of mechanized decision making. A large number of such decision-making devices are now in use, monitoring some physical process and organizing corrective or warning action. These devices are preferred to humans in these situations because the monitoring task they are doing is simple, they work quickly and because the decisions they have to make are simple and they do not fatigue. A human is still used where the decisions and actions required are complex and nonrepetitive and where the skills required are not precisely defined. A human may learn from his experience.

Recent advances in electrical technology have produced machines with a large high-speed, decision-making ability and a large quick access memory. These machines have been compared to human brains, and their abilities have been organized (by humans) so that they can play chess, for example, and defeat an unskilled human.

Other concurrent technical advances have produced a need for human-like abilities in situations in which it is either difficult or impossible or costly for men to survive for biological reasons. The first example of these was the nuclear "hot-lab". Man's dexterity was transmitted through mechanical or electrical connections (1). Other proposed activities which may require some projections of man's abilities are deep water commercial exploitation and extra-terrestrial work. It would be possible to do work on the ocean floor, or on Mars (say), using machines similar to those in use, while the human operator stayed on earth. There are, however, some serious drawbacks to this arrangement. Two of the most obvious are the difficulty and cost of providing adequate feedback to the operator, and the effect of transmission delay. The use of on-the-spot decision making machines should reduce the effect of these difficulties (2).

The cost of sending machinery to these exotic places is high, so that the machine should be as versatile as possible. In order to match this mechanical versatility, the decision-making ability of the machine should be flexible and adaptable. Ideally, to use the machine efficiently, a complex man-machine "language" must be provided so that the human can cease being an "operator" and become a "supervisor", issuing complex instructions to his remote "slave" confident that, if possible, they will be executed or, if not possible, the "slave" will request further instructions. The "slave" must have a "language" sufficiently rich to describe its world and the supervisor must organize his thoughts to correspond to this "language". This requires effort on the part of the human supervisor. "In this way a paradoxical situation arises: the richer the input language and the nearer the statement of the problem to living human language, the greater the labor the man must expend in matching his own original text to the capabilities of the formal system". (3)

The work done by a manipulator has a more human-like quality than that of a steam engine, say, or a Jacquard loom (4), to consider two examples of artifacts which have been "controlled" automatically for over a century. Ernst (5) in his thesis reports applause from an audience at one stage of a demonstration of his computer-controlled manipulator. His machine would stack blocks and put them into a box. If its environment changed, the box, for example, being moved, the machine would search for the box.

A machine that can be "led by the hand" and learn a set of motions is "Unimate" (6), which is a versatile material handling device.

The need for some form of supervisory control of manipulative devices was mentioned by several speakers at the Project ROSE Seminars in 1964 (7).

There is a continuous program of work on manipulation in the Man-Machine Systems Laboratory at M.I.T. These include several theses (8), (9) on the development of touch sensors. Ferrell (10) discovered that an information transmission delay did not make remote manipulation impossible, as had previously been surmised (11). Ferrell pointed out, however, that the time taken to complete a given task increases with delay. With a long transmission delay and a complex task to perform, this increased time-to-completion may be intolerable.

2. PROBLEM STATEMENT

This report describes a study of remote manipulation using a digital computer simulator.

The problems investigated were:

1) Is a real-time simulation of a manipulation task possible, and can useful experimental results be obtained from such a simulation?

2) Will the operator of a rate-controlled manipulator adopt a "move-and-wait" strategy when working with a transmission delay, as suggested by Ferrell (12) ?

3) What are the effects of a combination of transmission delay, inertial time constant, and intermittent display of a human operator's performance in a manipulative task?

4) Will supervisory control have any advantages in such a task?

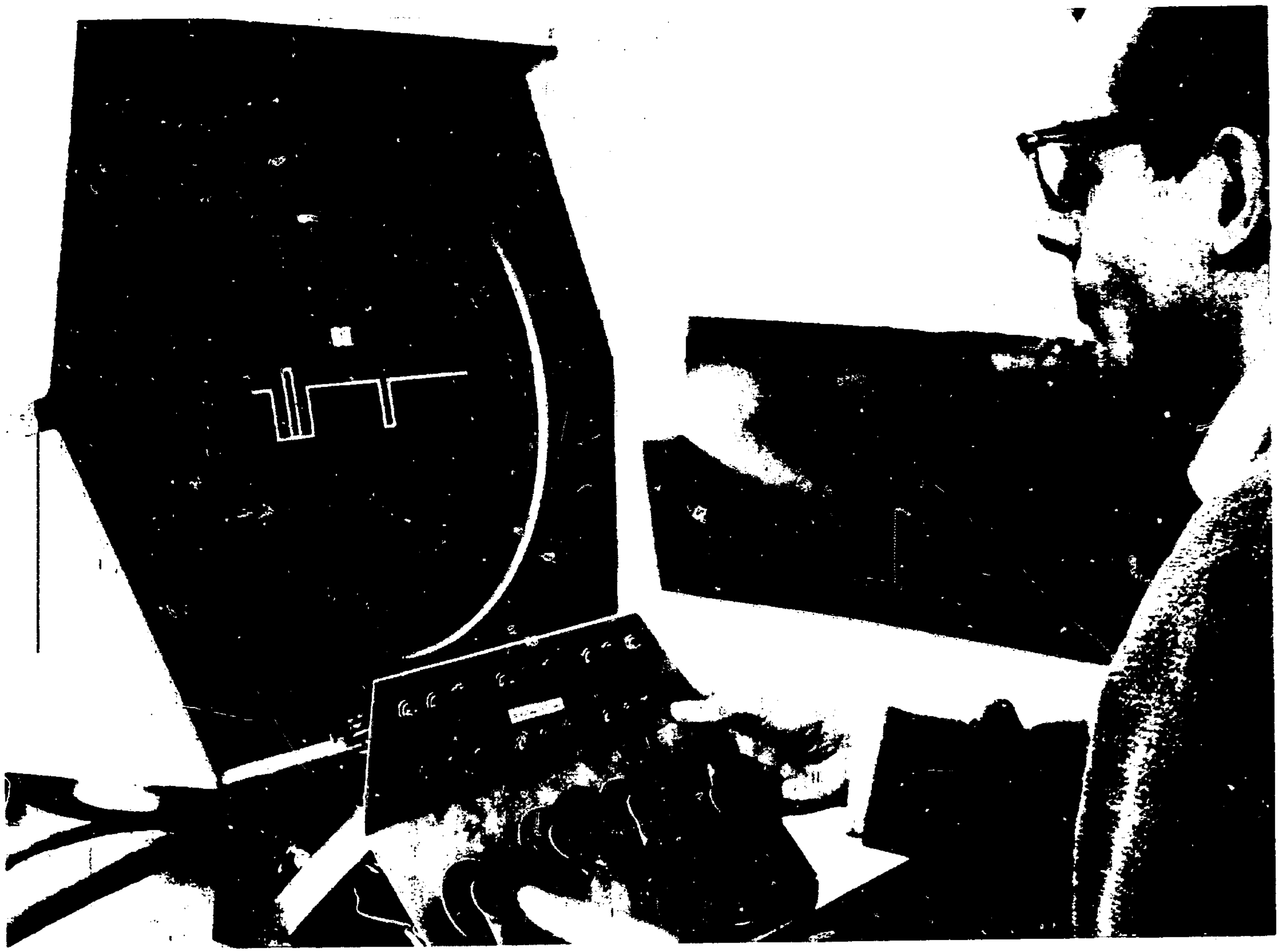


Fig. 1. View of the Display, and Control Panel

PRECEDING PAGE BLANK NOT FILMED

3. JUSTIFICATION FOR THE EXPERIMENTAL PROCEDURE

The PDP-1 Computer of the Research Laboratory of Electronics at M.I.T. (see Appendix I) was available for these experiments. It has a speed high enough for the purpose and a cathode ray tube display. A real-time experiment requires that the computer be "available" to the user as he requires. A display requires considerable computational effort if it is to seem real to the user. The PDP-1 computer was adequate for the purpose provided that the task to be simulated was simple.

Making the task simple also made the evaluation of the results easier.

The task was two-dimensional throughout. The operator had control over a pair of fingers using the switch panel which is one of the in-out facilities of the computer. Using these fingers, the operator can grasp a rectangular block and move it across the screen. He is required to place it in a target hole (Fig. 2).

Rate-control of the manipulator was obtained using knob operated potentiometers available on the switch panel.

The measurements that can be made from such an experiment include the time taken to complete the task and the fuel consumed in performing it. Another measurement is the amount of telemetry

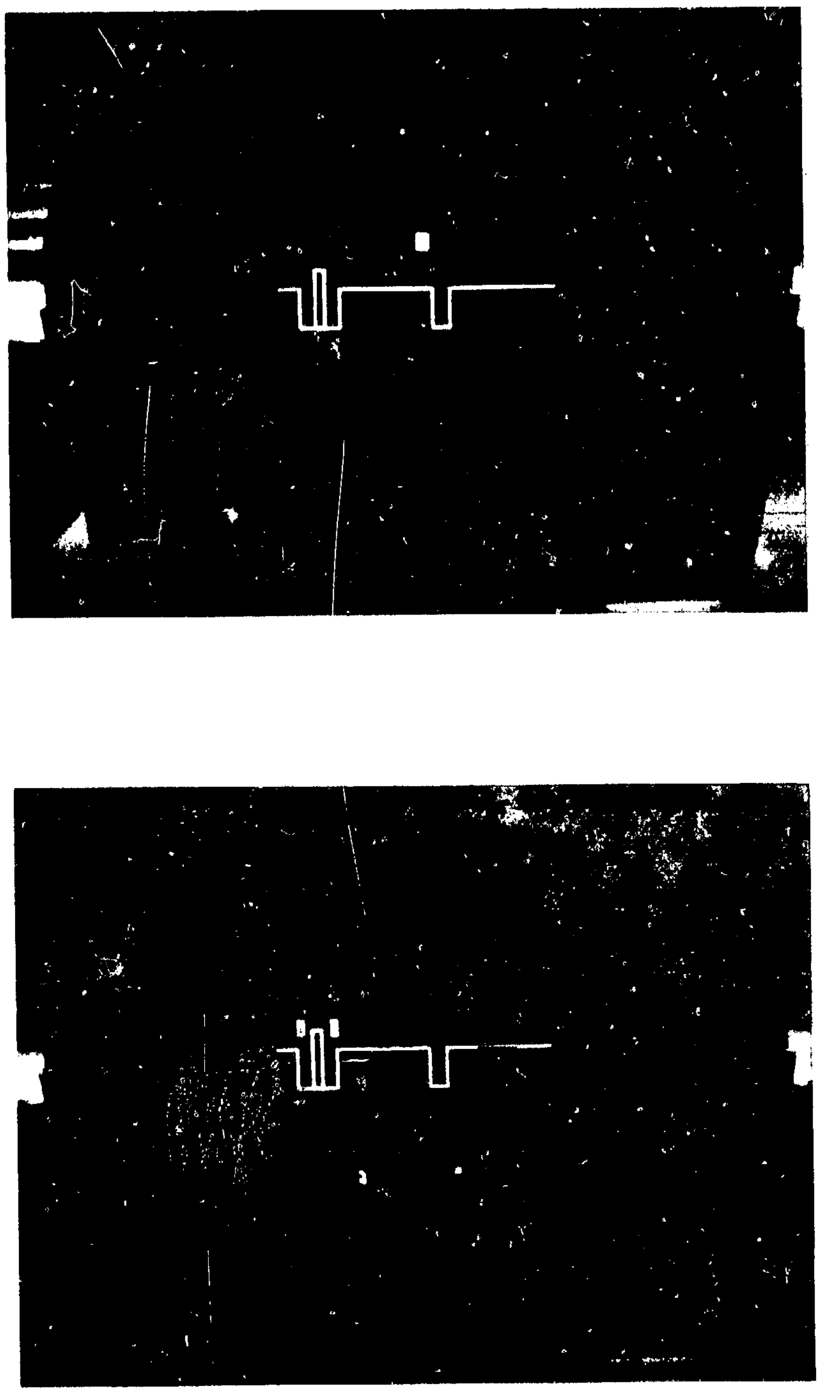


Fig.2(a). An experimental run, showing the start (top), and the jaws positioned to grasp the block (bottom).

Fig.2(b). The block being raised (top), and finally positioned (bottom).

required. It is easy to measure the time required using a computer with a stable cycle rate. It is fairly easy to measure energy or momentum expended provided that there are no impacts or rebounds during the experiment. When there are, then the method used to simulate them may affect the validity of the fuel or energy measured.

The use of supervisory control should reduce the mental effort and attention that the task requires from the man. A measure of this attention was obtained by counting the number of control signals that the operator "sent" to the remote manipulator. Supervisory control should reduce the number of these.

With such a simple task, it is possible to preprogram the entire job. This would not provide any significant results. In order to stimulate interaction between the operator, the remote controller and the manipulator (Figs. 3 and 4), three preprogrammed logical sets of instructions were provided which the operator could use at his discretion. These would use information from touch sensors, or from a timing device. These sets of instructions were chosen to perform the operations requiring precise monitoring and actuation by the operator.

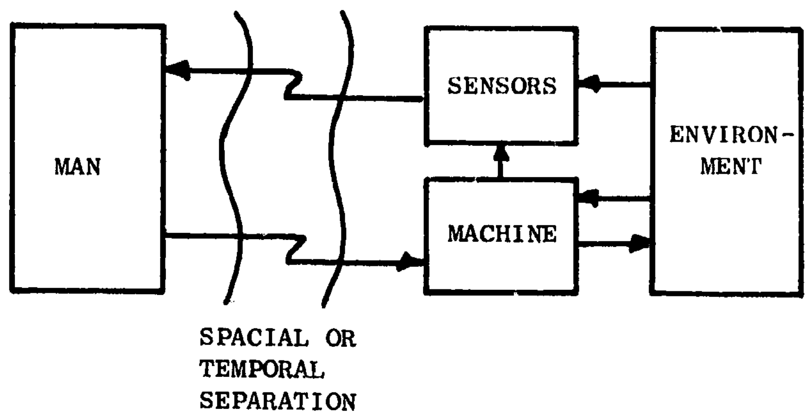


Fig. 3. Man-Machine Interactions.

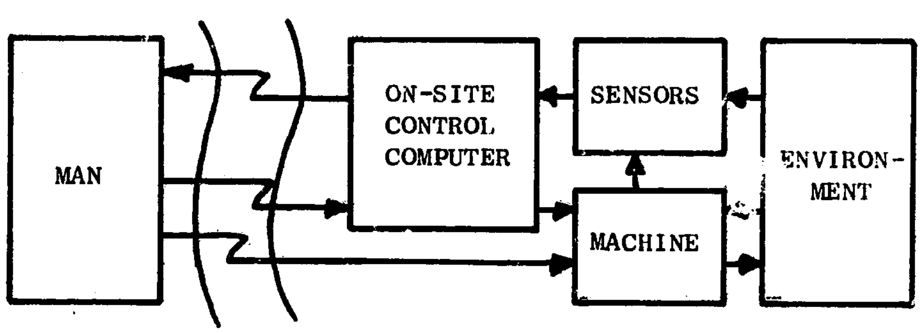


Fig. 4. Man-Computer-Machine Interactions.

4. THE SIMULATION PROGRAM

4.1 General Description.

The program occupies nearly four thousand words of the core memory of the PDP-1 Computer. About eight hundred of these are storage tables. The program is divided into separate parts which are described in the following sections of this chapter.

During an experimental run, the program runs in a loop (Fig. 5) through all sections, except the Output and Reset parts. The cycling rate is 15 per second. This produces a slight flicker on the screen, but the task appears continuous to the subject.

4.2 The Simulated Manipulation Task and the Manipulator.

Displayed on the Cathode Ray Tube Display Screen in front of the subject are a block, a pair of fingers and a ground profile with two holes in it. The block may be moved, grasped or knocked over by the fingers, and will fall to the ground if released above it. If the fingers try to push the block into the ground, or if the block has too high a vertical velocity when it hits the ground, then the program stops, prints an error statement, and resets itself to the initial positions. If the block when knocked over rotates

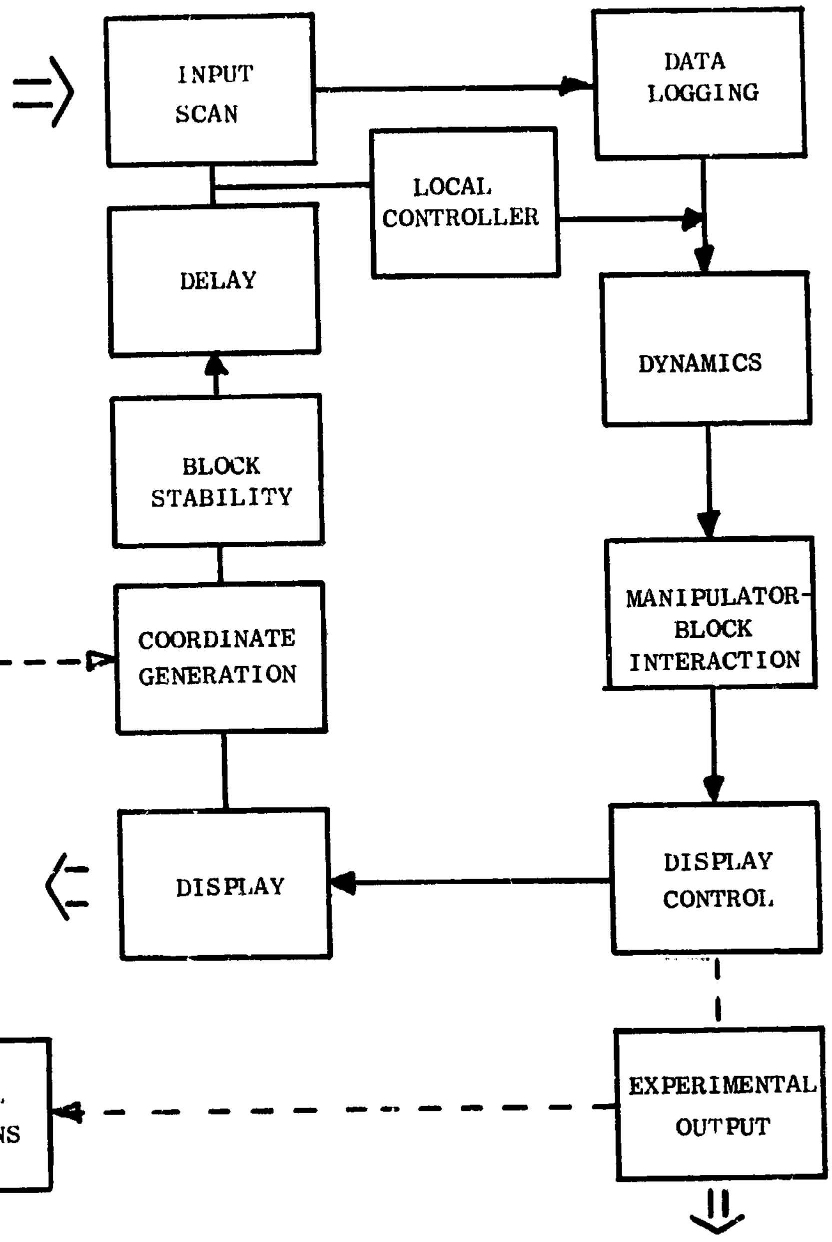


Fig. 5. Main Structure of Program.

through a right angle, an error statement is printed and the program restarted.

The features of the display are all rectangular, and the coordinates of their corners are stored and updated by the program. The interactions, such as pushing, grasping and bouncing, are effected by comparing all these coordinates and determining whether any of them overlap, and making appropriate decisions about the motions of the block and the jaws. Figure 6 is a simplified flow chart of the part of the program which decides whether the block should fall, bounce or topple sideways. When the fingers, or jaws, overlap the block on both sides, the block is defined as grasped, and movements of the jaws move the block, but with the mass and weight of the block added. If the fingers meet through the block, an error statement is printed because the fingers are said to have crushed the block. If the fingers touch the ground, an error statement is printed out.

4.3 The Local Controller.

Three particular sequences of operations have been preprogrammed, and simulate the logical decision making processes that an on-site controller might perform. They are arranged to carry out the three precise actions that the subject has to perform: picking up the block, finding the narrow hole into which the block should be placed, and lowering the block to the ground and releasing it.

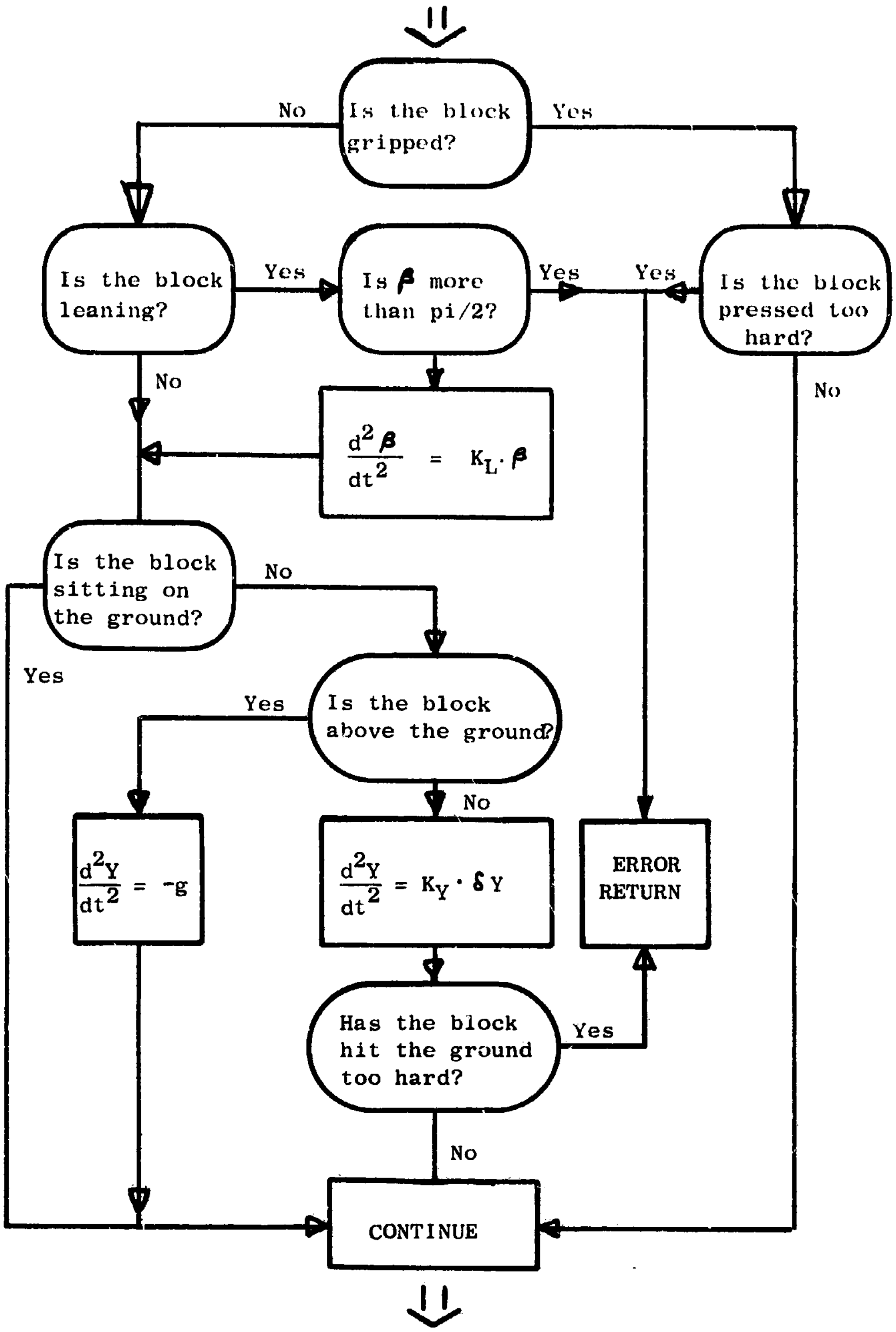


Fig.6.Flow Diagram of Block Stability Section.

These subroutines require signals from on-off touch sensors on the inside face of the fingers, a load sensitive sensor to determine whether the block is resting on the ground, and velocity sensors in the horizontal (x) direction and the vertical (y) direction.

While the subroutines are "switched on", the direct instructions from the subject to the task are ignored. The subject may switch off the subroutine when he wishes. The subroutine returns control to the subject when it has completed its set of instructions. The logical arrangement of these subroutines is detailed in Figures 7, 8 and 9.

4.4 The Input Section.

The subject moves the fingers and issues instructions using a bank of 18 switches and two knob-driven potentiometers (Fig. 10). These are "read" by the program on each cycle, the output of the potentiometers passing through an analog to a digital converter, and then shifted right to prevent contact noise being read by the program. The switches are read as a complete word into the computer. The program stores this word and interrogates appropriate sections of this word during the cycle. The switches enable the subject to open and close the jaws, to demand vision, to start and stop the experimental run, and to transfer control to the on-site subroutines. The knobs provide a velocity input to the fingers. If the fingers are grasping

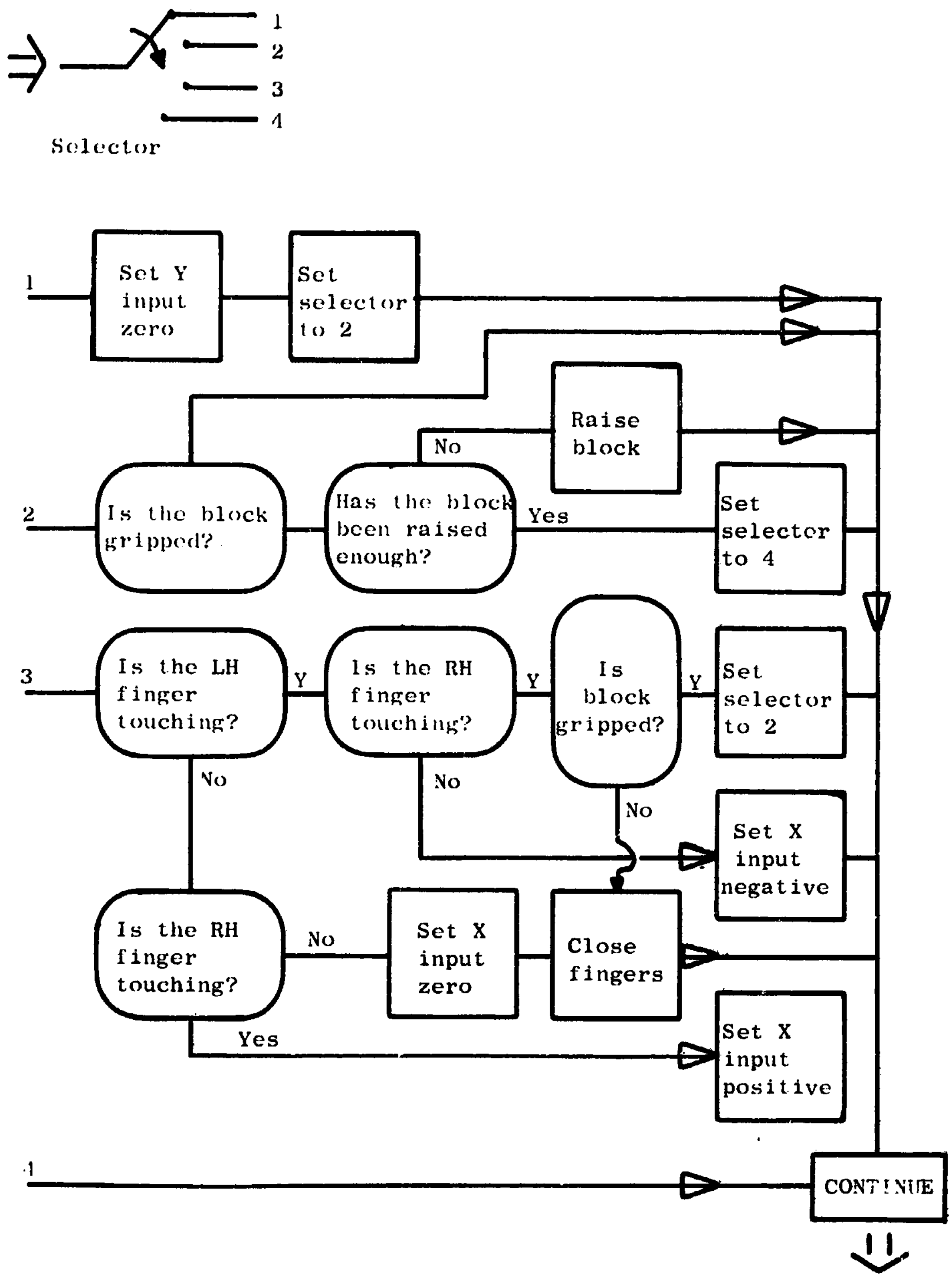


Fig. 7. Flow Diagram of Grasp and Lift Subroutine.

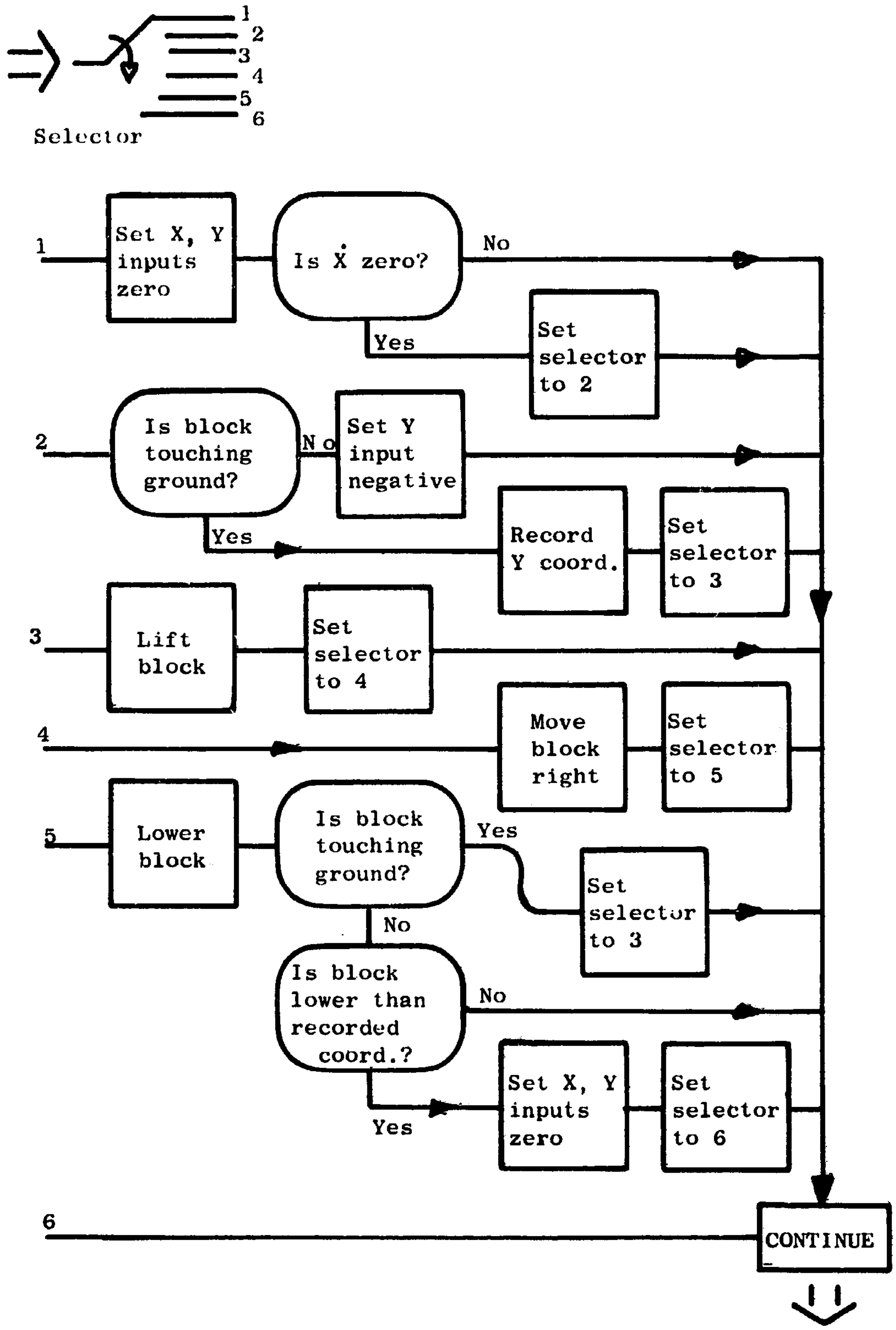


Fig. 8. Flow Diagram of Lower and Search Subroutine.

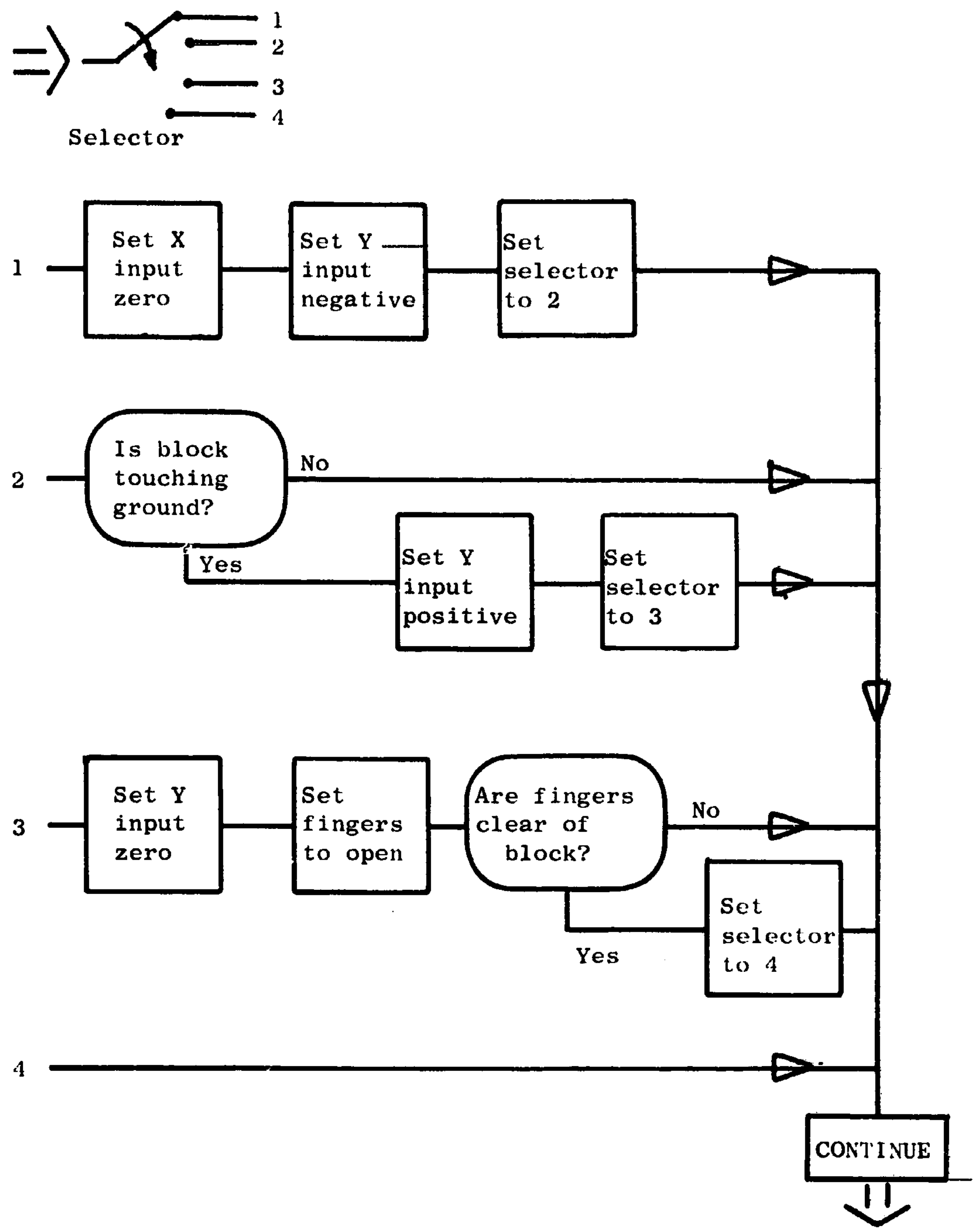


Fig. 9. Flow Diagram of Lower and Release Subroutine.

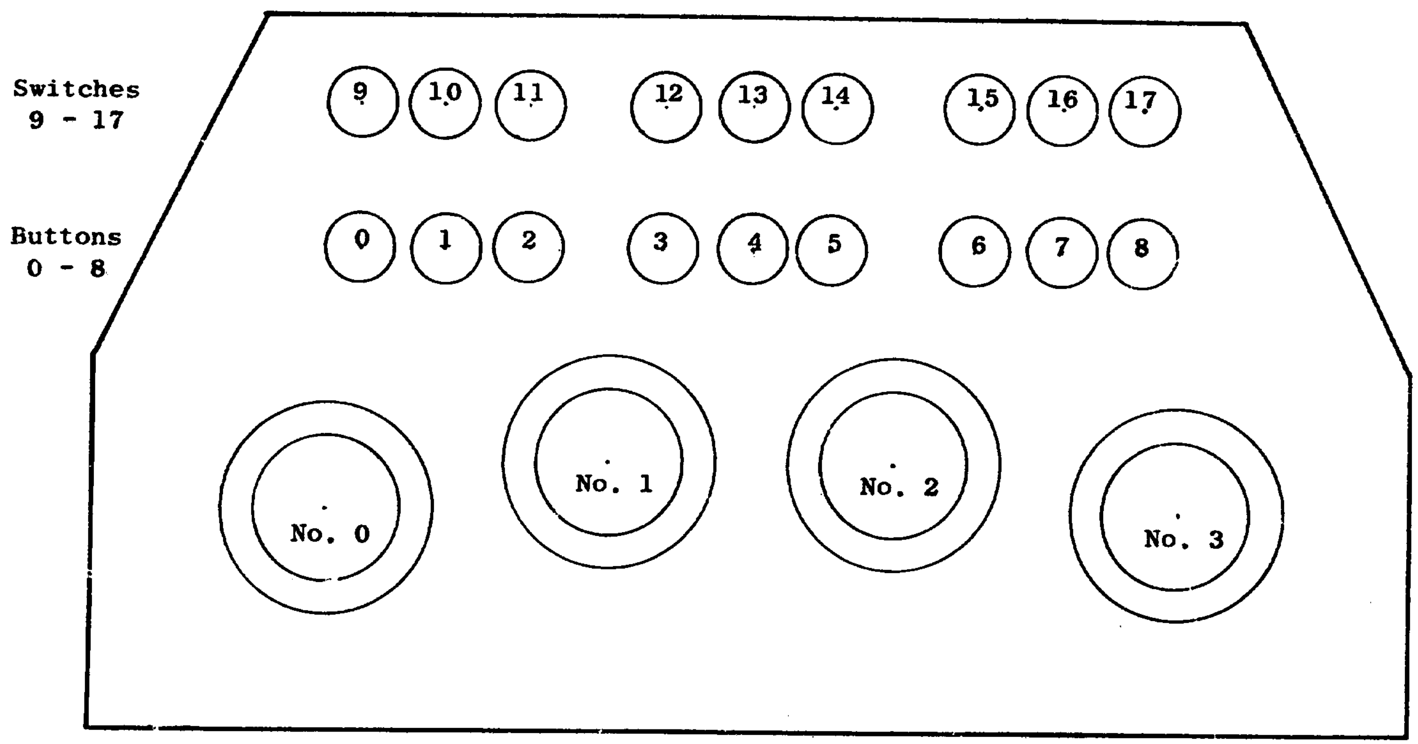


Fig. 10. Control Knobs and Switches (half-scale).

the block, this input is passed through a first order exponential lag whose time constant depends on the mass of the block, and may be varied from 1/8 to 8 seconds, by the experimenter (Fig. 11). A gravity term is added to the equation of motion for the y direction, but does not affect the null point of the control knob. When the time constant (due to the mass) is 3.14 seconds, the gravity term alters the y velocity by 25%.

4.5 The Display Control and Coordinate Transfer Section.

This section allows the experimenter to present various types of display to the subject, or to arrange for the subject to demand vision when he requires it (Fig. 12). The program maintains two sets of coordinates for the fingers and the jaws (but one set for the ground). Of the two sets, one is the "real" set of coordinates which determine the interactions, overlap, pushing, etc., while the "imaginary" one is the displayed set. The complete display is built up from the corners of the figures, using a line generating subroutine. It is possible to have differing types of display. Among these are the continuous, continually updated display (a TV-like image), an interrupted TV-like display, or a display of "static" pictures, obtained by transferring the "real" coordinates to the "imaginary" coordinate registers at large time intervals. It is also possible to have a display only when the subject demands it through his control panel.

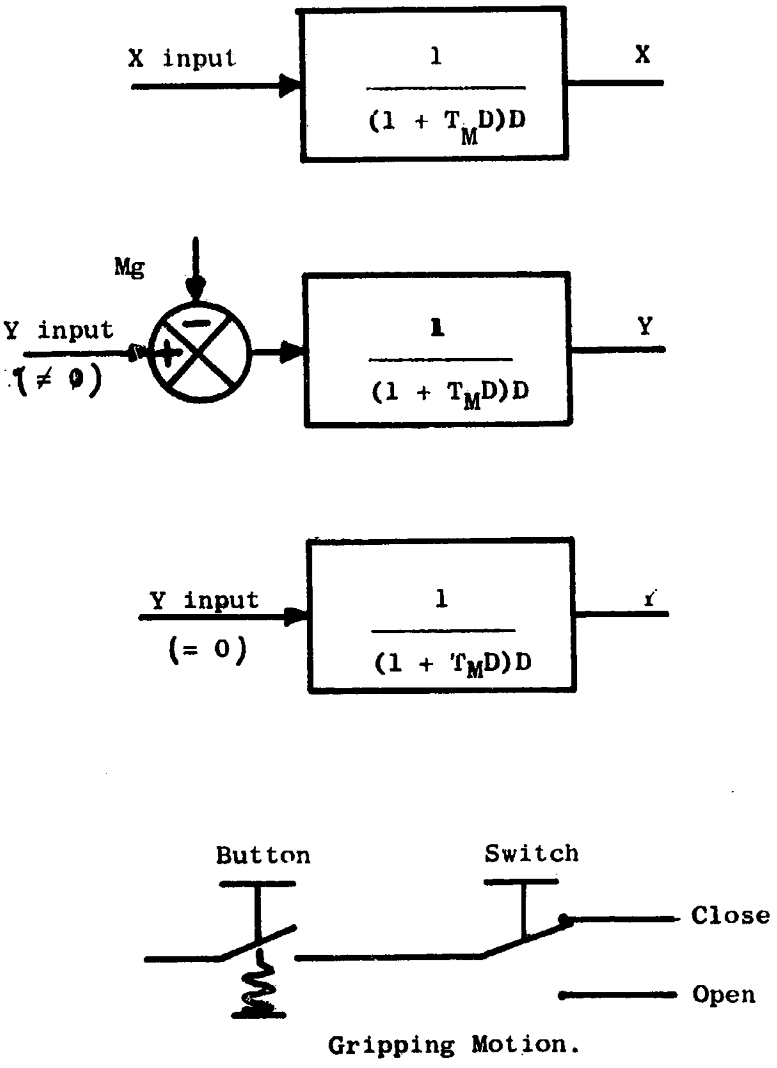


Fig. 11. Block Diagram of System Dynamics.

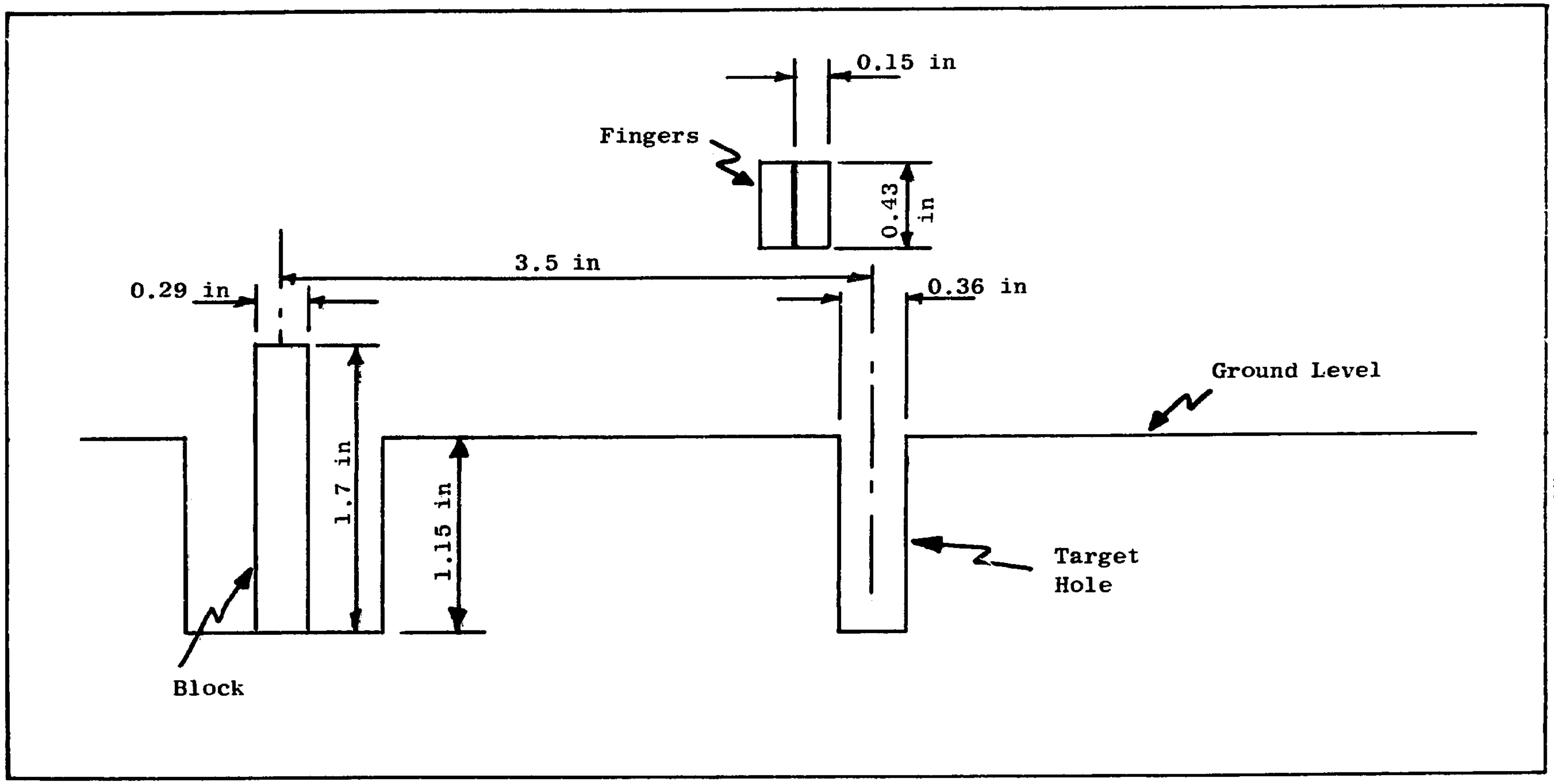


Fig. 12. The Display.

4.6 The Data Logging Section.

This section of the program records the commands given by the subject through the control knobs and switches. On each program cycle, the settings of these control knobs and switches is compared to those recorded on the previous cycle. Any change is noted, together with the time of occurrence and stored in a table in coded form. For details of the coding, see the instructions for using the program.

4.7 The Delay Section.

This section lets the experimenter introduce a delay between the operator and the system. The maximum delay possible is 17 seconds, in steps of 1/15 second. Three sets of memory registers store the switch input signal and the digitally converted x and y input signals. The signals are read into and out of these storage registers by indirect addresses which are indexed each cycle. At the start of each experimental run, the spacing between these indirect addresses is set equal to the contents of register dly.

4.8 The Output Section.

At the end of each experimental run, the results are printed and punched out (see Fig. 13). The time taken is computed by dividing the number of program cyclés by the time per cycle. The number

```
5/7th 1966  Type 60D30  Subject DJB Run 014707
Total time secs 000261
Trans pictures 002367
Commands 000030
Commands 000030
Fuel 011704
Error-free run
Distance moved 012230
Score  000000
Display cycle 4.25 secs
Display interval 4.25 secs
Delay 3.14 secs
Time constant 10.42 secs
Task distance 060000
Tolerance 001000
phs 001000
Punching completed
```

Fig. 13a. Example of Program Type-Out.

```
run,                 014707
dte,                 000007
tab,                 450000
          050000
          320000
          220005
          430021
          210030
          450033
          310042
          040046
          320207
          310216
          220220
          320277
          310302
          050312
          020313
          050423
          010424
          320453
          310457
          320457
          310457
          410522
          450535
```

Fig. 13b. Example of Program Punch-Out.

of times that the "real" coordinates have been transferred to the "imaginary" display ones is printed. The number of commands (computed by the Data Logging Section) is printed, as is distance traveled, and "fuel" used. The "fuel" consumption is proportional to the modulus of the momentum change occurring during one run. The "score" or "cost" is computed, determined by the costs attached by the experimenter to time, fuel, commands and number of display coordinate transfers made.

4.9 The Reset Section.

This section clears all the storage tables, resets all counters to zero, and assigns the knobs to the program.

5. EXPERIMENTAL PROCEDURE AND RESULTS

5.1 Conduct of the Experiments.

The experiments were divided into two phases. Four subjects, all male graduate engineering students between 23 and 31 years old, were used, three of them during the first phase of the experiments and three during the second phase. One of these subjects was the author.

The instructions to the subjects were the same for every run. They were to move the displayed block into the small displayed hole using the fingers displayed on the screen. They should then close the fingers and press the finish button. They were to use the least number of commands that they could. They should avoid making errors. The subjects were told that their time to complete the task would be measured, but that they were not to try to complete the task as quickly as possible.

5.2 Effect of Delay, Dynamic Lag and Intermittent Vision on Performance.

During Phase I of the experiment, the three subjects each completed 10 consecutive experimental runs of the task under 18 different experimental conditions. These conditions were a factorial arrangement of 3 different time constants, 3 different delays, and

2 different display conditions. The time constants were 0.27 seconds, 1.0 second, and 8.5 seconds; the delays were 0.0 seconds, 0.27 seconds, and 3.2 seconds; and the display conditions were (a) a continuous view, and (b) an interrupted view lasting 0.6 seconds at intervals of 4.3 seconds. The subjects had received considerable training with all the different experimental conditions before data were taken.

During this phase of the experiment, the maximum velocity in the x and y directions was 0.23 inches/second. Two of the restrictions were also absent. The fingers had no restriction as to the squeeze they could exert on the block, and they were permitted to touch the ground.

The somewhat arbitrary arrangement of knobs and switches caused the subjects some trouble during the initial part of their training period. The difficulty appeared to be remembering which switch controlled which motion. After about four hours work, the subjects appeared to have adjusted themselves to the arrangement of the controls. A similar adjustment was noted by Crawford (13). In his experiments, the subjects' performance after practice while using lever controls approached their performance using a joystick controller.

Two subjects were unable to detect the 0.27 second delay, declaring that they did not believe that there was in fact a delay. Initially several errors occurred because the subjects could not distinguish between the effects of "mass" (time constant) and delay. The 1.0 second time constant did not seem to cause any difficulties to the subjects, but the 8.5 second time constant caused difficulties and was considered difficult at first. With this time constant, the gravity field exerted a large force on the block and jaws combined. This altered the maximum speed attainable in the vertical direction by 25%, thus making the ratio between upward and downward speeds 75/125, i.e., 0.6:1.0. The effect of intermittent vision varied between subjects. One (the author) disliked this intensely, and his performance was impaired, while the other two appeared to adjust reasonably well to this condition although considering it harder.

People watching the experiment for the first time ask whether subjects try to "trade-off" the number of commands (the variable of interest) against time, which is supposedly of no interest. A policy of moving exceedingly slowly should enable a subject to avoid any miscalculations and thereby perform the task with a minimum of commands.

During their first few runs most subjects try this, but they abandon this policy soon, as they succumb to a desire to "get things done". The subjects in general used the maximum velocity available, except when putting the block down. For several runs the subjects tend to use the controls continuously as they try to "steer" the objects. After some runs with an apparent delay, however, the subjects adapted the "move and wait" strategy noticed by Ferrell (10).

(Classical instability was noticed once when an inexperienced operator, not a regular subject, was attempting to maneuver the block with a 3.2 second delay and 8.5 second time constant.)

The subjects also tended to avoid controlling the x and y motions simultaneously. At the end of this set of experiments, the task had become a repetitive and easy one with little interest for the subjects.

The results of Phase I of the experiment are shown in Tables 1, 2 and 3 where the results for each subject and each experimental condition are listed. The listed figures are the mean and standard deviation (corrected for small samples) of 10 good experimental runs, obtained when the subjects behavior had stabilized. Also listed are the mean and standard deviation obtained by summing over the three subjects. The average number of commands used by the three subjects is plotted against delay in Figure 14 for continuous vision, Figure 15 for intermittent vision, Figure 16 for both combined.

COMMANDS

Time Constant (secs)	Vision	Delay (secs)	Subject B.F.	Subject D.B.	Subject S.M.	3 Subjects Combined
0.27	C	0.0	18.7 (1.7)	18.7 (1.3)	19.7 (4.7)	19.1 (3.1)
		0.27	18.6 (1.2)	18.4 (2.5)	17.4 (1.5)	18.1 (1.8)
		3.2	19.1 (2.2)	21.4 (2.0)	20.2 (2.1)	20.2 (2.2)
	I	0.0	19.2 (1.7)	22.4 (3.6)	17.8 (2.1)	19.8 (3.1)
		0.27	16.6 (1.6)	19.6 (3.2)	20.1 (1.7)	18.7 (2.8)
		3.2	17.1 (1.4)	22.7 (1.3)	22.8 (3.0)	20.9 (3.3)
1.0	C	0.0	18.0 (1.4)	20.7 (2.8)	17.2 (1.7)	18.6 (2.5)
		0.27	19.4 (1.9)	17.3 (1.8)	17.0 (1.8)	17.9 (2.0)
		3.2	22.2 (2.4)	20.2 (3.5)	16.0 (1.3)	19.5 (3.8)
	I	0.0	17.2 (2.9)	21.3 (1.9)	18.6 (2.2)	19.4 (2.8)
		0.27	17.5 (2.3)	23.0 (2.7)	19.0 (2.3)	19.8 (3.3)
		3.2	19.1 (2.4)	23.6 (1.5)	19.7 (1.5)	20.8 (2.7)
8.5	C	0.0	17.3 (1.3)	14.9 (1.3)	16.9 (2.0)	16.4 (1.8)
		0.27	19.4 (1.8)	15.7 (1.7)	16.5 (1.7)	17.2 (2.5)
		3.2	16.2 (2.8)	17.4 (1.7)	20.8 (3.2)	18.1 (3.2)
	I	0.0	19.2 (1.4)	19.1 (2.0)	20.7 (3.4)	19.6 (2.4)
		0.27	20.6 (3.2)	19.4 (3.0)	24.1 (2.5)	21.4 (3.5)
		3.2	21.2 (2.4)	21.4 (3.1)	22.0 (3.2)	21.5 (2.8)

Note 1. C indicates continuous visual display.
2. I indicates intermittent visual display.

The tabulated figures are the mean and standard deviation for 10 completed experimental runs for each subject. The standard deviations have been corrected for small sample size.

Table 1. Effect of Delay, Lag and Intermittent Vision on Number of Commands.

TIME
(Seconds)

Time Constant (secs)	Vision	Delay (secs)	Subject			3 Subjects
			B.F.	D.B.	S.M.	Combined
0.27	C	0.0	62.8 (6.4)	65.9 (3.6)	54.7 (4.1)	61.1 (6.6)
		0.27	64.2 (4.4)	89.6 (8.7)	61.5 (5.4)	71.0 (14.0)
		3.2	96.2 (9.0)	93.1 (10.4)	86.0 (3.5)	90.0 (10.0)
	I	0.0	87.7 (8.4)	86.1 (4.7)	67.1 (3.3)	80.3 (10.9)
		0.27	88.2 (5.0)	86.4 (1.6)	82.6 (6.6)	85.7 (5.3)
		3.2	118.5 (7.0)	130.6 (8.0)	93.6 (6.3)	114.2 (16.7)
1.0	C	0.0	61.4 (3.5)	70.4 (7.3)	57.5 (2.7)	64.0 (7.0)
		0.27	74.3 (7.1)	64.7 (3.4)	59.4 (3.7)	66.1 (7.7)
		3.2	108.5 (10.5)	95.9 (10.1)	71.2 (4.6)	91.9 (17.5)
	I	0.0	92.6 (13.6)	96.1 (3.3)	76.5 (8.3)	88.5 (12.7)
		0.27	105.7 (11.3)	101.5 (11.5)	78.3 (4.5)	95.1 (15.1)
		3.2	140.7 (11.4)	134.3 (4.5)	116.5 (8.3)	130.5 (13.4)
8.5	C	0.0	102.5 (14.1)	75.5 (3.7)	78.6 (11.5)	85.5 (16.1)
		0.27	106.0 (10.3)	86.0 (2.7)	77.2 (8.9)	89.7 (14.3)
		3.2	134.1 (24.1)	100.9 (6.4)	115.0 (17.0)	116.6 (21.9)
	I	0.0	112.3 (11.1)	91.7 (4.0)	108.5 (13.1)	104.1 (13.1)
		0.27	122.3 (14.1)	95.5 (17.9)	120.0 (10.4)	112.6 (18.4)
		3.2	152.9 (11.1)	116.7 (8.0)	142.3 (12.1)	137.3 (18.3)

Note 1. C indicates continuous visual display.
2. I indicates intermittent visual display.

The tabulated figures are the mean and standard deviation for 10 completed experimental runs for each subject. The standard deviations have been corrected for small sample size.

Table 2. Effect of Delay, Lag and Intermittent Vision on Completion Time.

FUEL

(Arbitrary Units)

Time Constant (secs)	Vision	Delay (secs)	Subject B.F.	Subject D.B.	Subject S.M.	3 Subjects Combined
0.27	C	0.0	261 (12)	275 (49)	321 (62)	287 (51)
		0.27	265 (16)	170 (25)	293 (11)	242 (56)
		3.2	289 (52)	259 (34)	635 (17)	394 (64)
	I	0.0	235 (10)	243 (13)	337 (57)	271 (57)
		0.27	230 (9)	173 (37)	362 (39)	255 (85)
		3.2	518 (3)	528 (24)	390 (58)	479 (72)
1.0	C	0.0	688 (50)	660 (84)	763 (38)	704 (71)
		0.27	623 (38)	588 (77)	758 (45)	656 (89)
		3.2	621 (21)	843 (121)	886 (91)	783 (143)
	I	0.0	588 (35)	578 (77)	779 (67)	649 (101)
		0.27	556 (35)	523 (13)	879 (100)	652 (174)
		3.2	869 (19)	900 (124)	1026 (169)	932 (137)
8.5	C	0.0	3600 (102)	4604 (448)	4415 (66)	4206 (509)
		0.27	3880 (77)	4412 (441)	4288 (66)	4193 (340)
		3.2	4042 (151)	4970 (562)	4690 (172)	4567 (519)
	I	0.0	3612 (13)	4027 (111)	4000 (44)	3888 (210)
		0.27	3842 (565)	4490 (726)	4342 (268)	4225 (603)
		3.2	3829 (59)	4363 (241)	4316 (242)	4169 (310)

Note 1. C indicates continuous visual display.
2. I indicates intermittent visual display.

The tabulated figures are the mean and standard deviation for 10 completed experimental runs for each subject. The standard deviations have been corrected for small sample size.

Table 3. Effect of Delay, Lag and Intermittent Vision on Fuel Used.

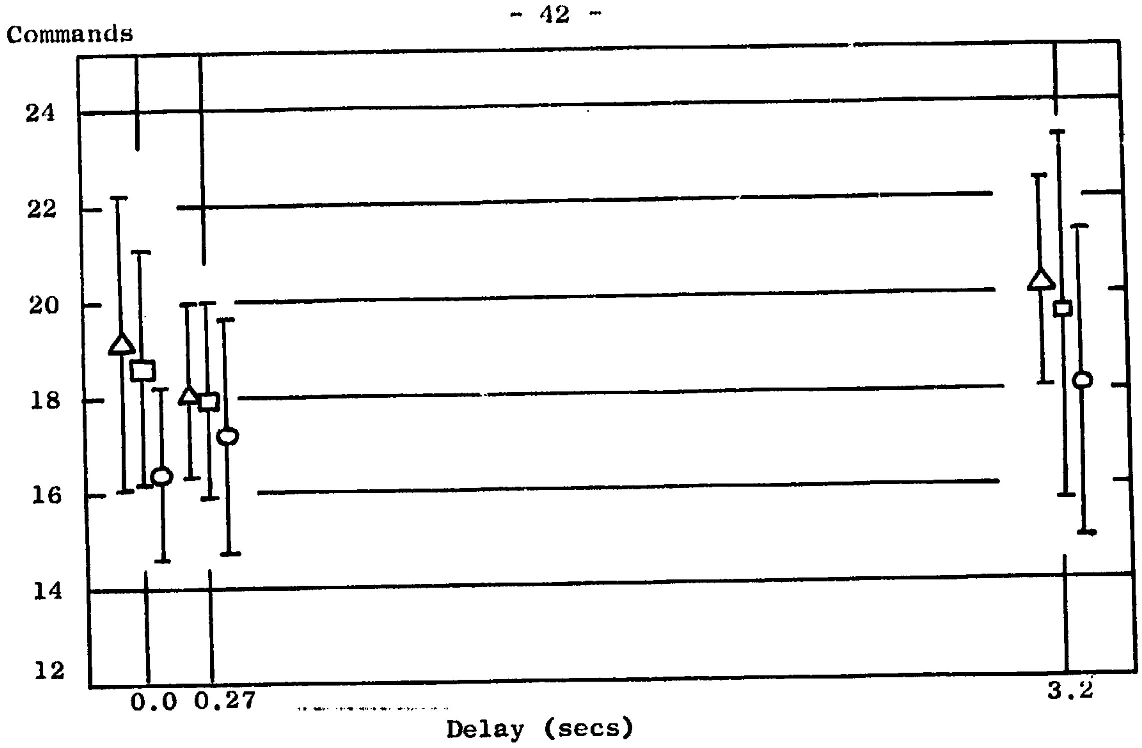


Fig. 14. Effect of Delay and Lag on Number of Commands used (Continuous Vision).

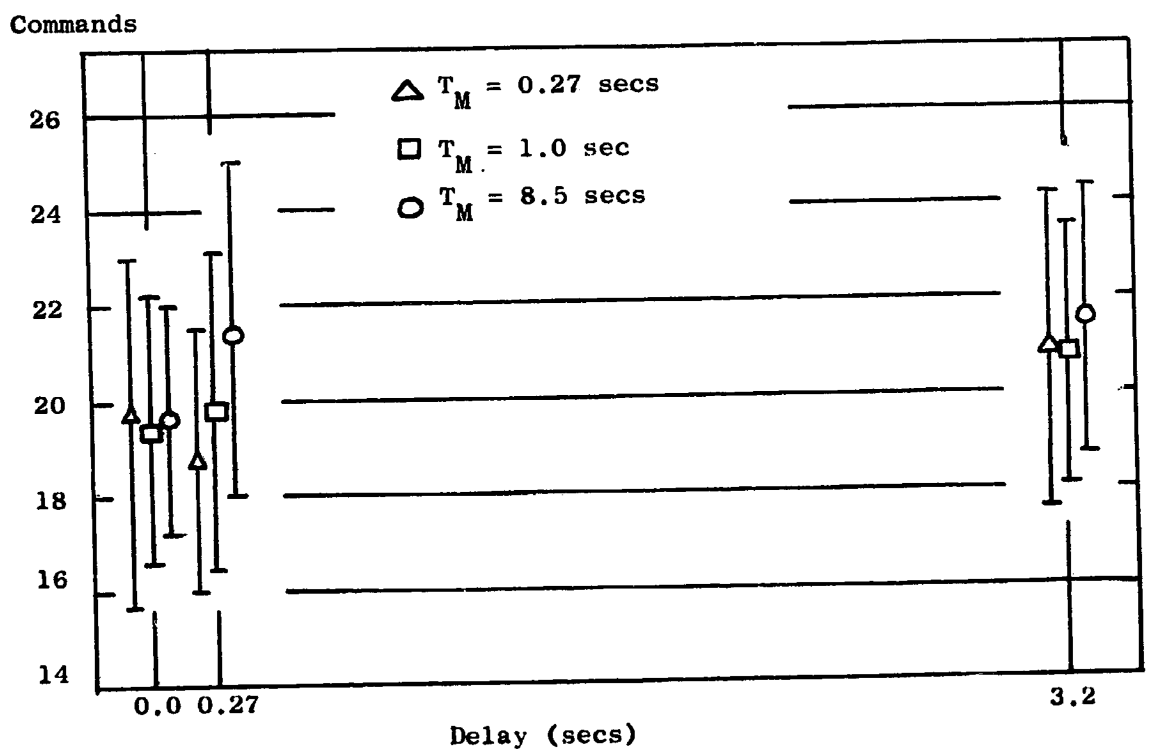


Fig. 15. Effect of Delay and Lag on Number of Commands used (Intermittent Vision).

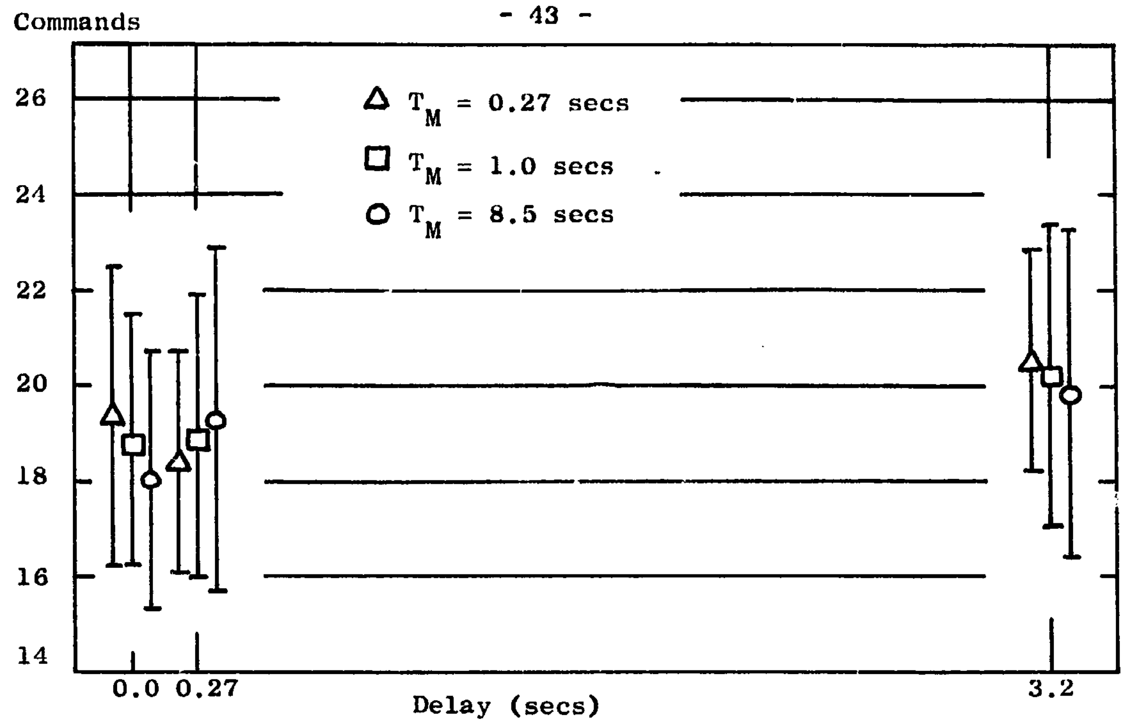


Fig. 16. Effect of Delay and Lag on Number of Commands used (Continuous and Intermittent Vision Combined).

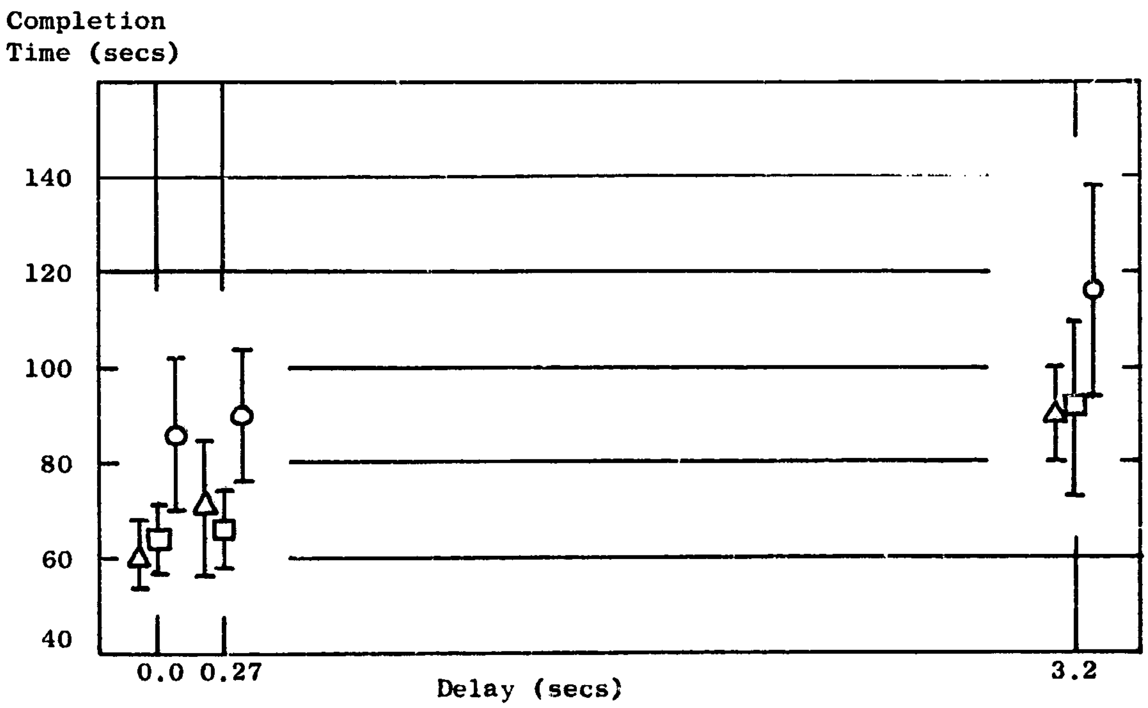


Fig. 17. Effect of Delay and Lag on Completion Time (Continuous Vision).

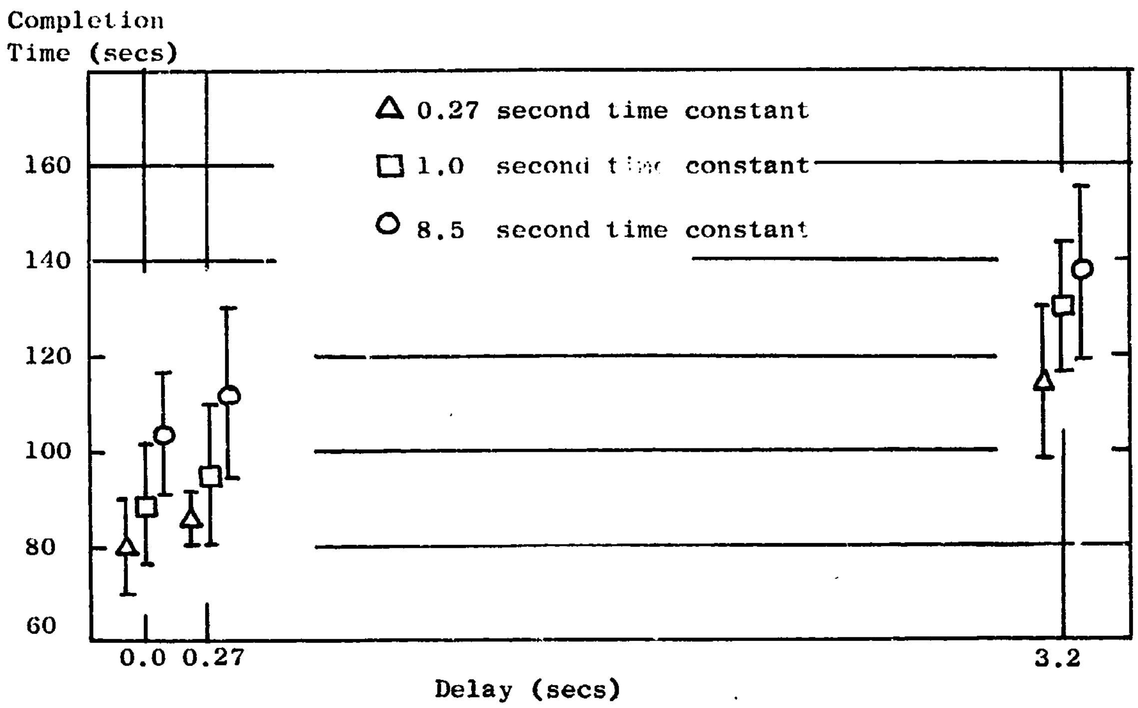


Fig. 18. Effect of Delay and Lag on Completion Time
(Intermittent Vision).

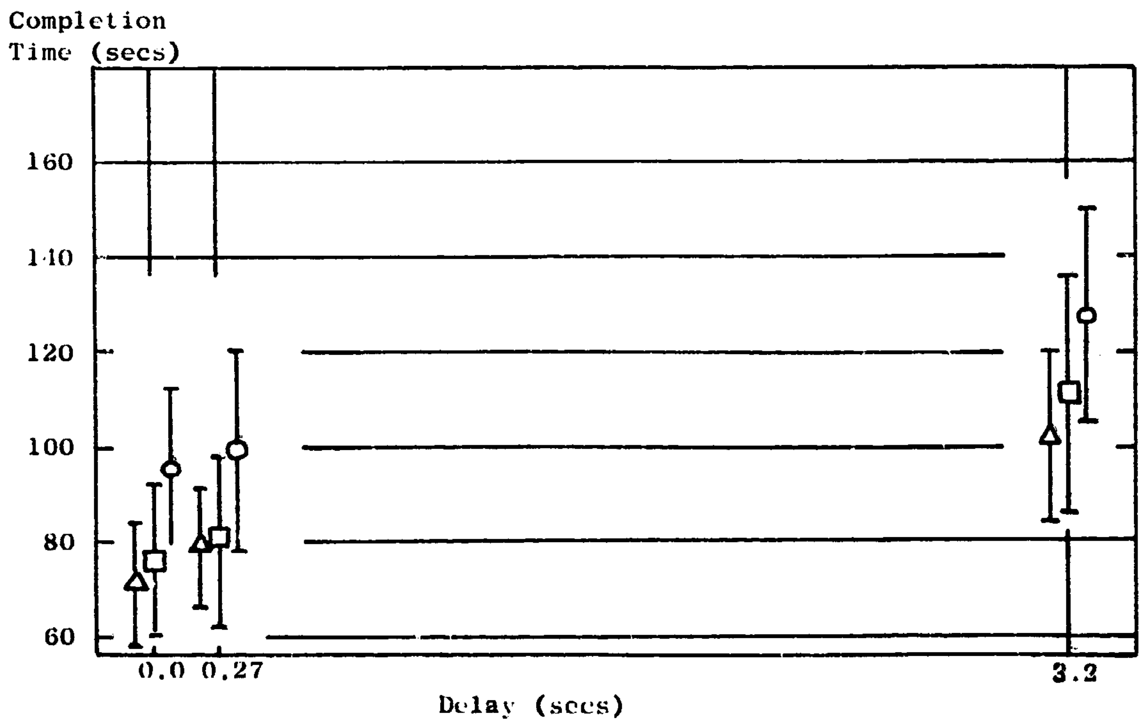


Fig. 19. Effect of Delay and Lag on Completion Time
(Continuous and Intermittent Vision Combined).

It is difficult to draw any clear conclusions from these results. Some trends may be noted, however. For continuous vision, the larger mass (time constant) required fewer commands than the smaller ones. One possible reason for this is that the larger mass required more concentration, whereas the smaller one, being easy, imposed no evident penalties on sloppy performance by the operator.

The times taken under varying conditions are plotted in Figures 17, 18 and 19. These show that increasing delay time and increasing the time constant both result in increased time to completion. The time taken increases linearly with delay, as was found by Ferrell (10). No noticeable difference occurs between the completion time for the two smaller time constants, when the subjects had a continuous visual display (Fig. 17), but with an intermittent display, the times for the three time constants are notably different (Fig. 18).

The "fuel" used while completing the task is plotted in Figures 20, 21 and 22. The data in these figures has been normalized by dividing by the amount used for no delay, 0.27 second time constant, and continuous display. The purpose of this normalization was to permit a comparison of the effects of delay and intermittent vision. A large mass will evidently require more "fuel" than a small one if moved in the same manner between the same two points.

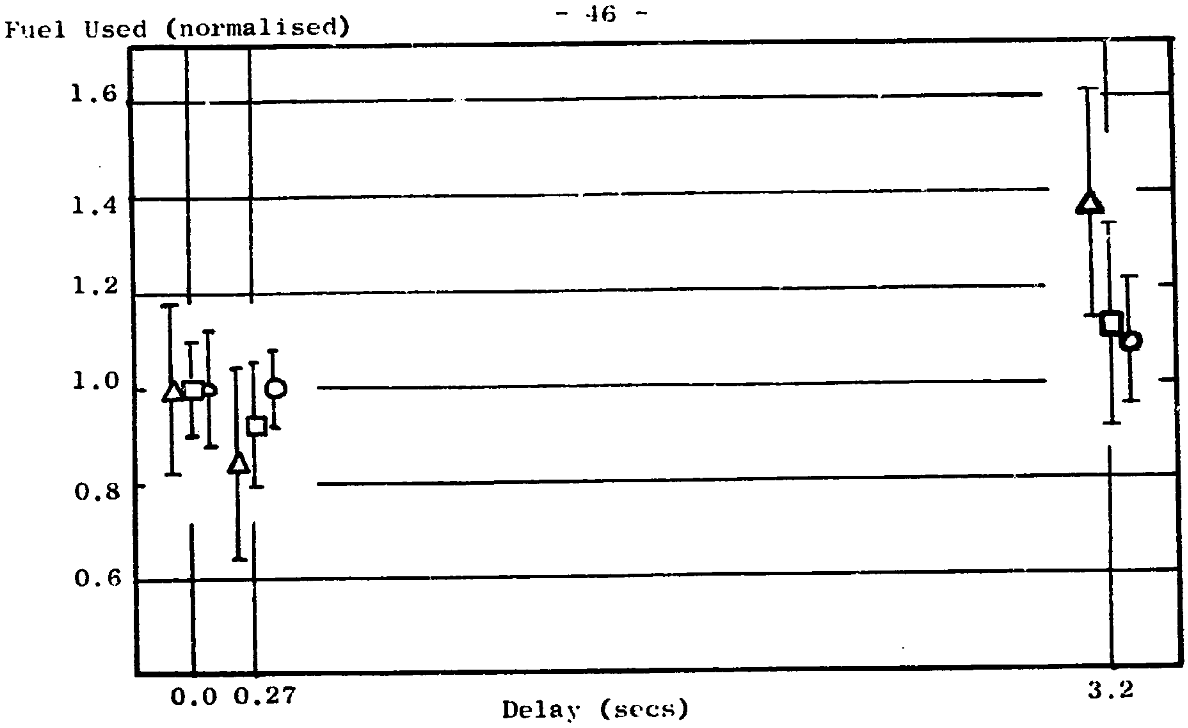


Fig. 20. Effect of Delay and "Delay plus Lag" on Fuel used (Continuous Vision).

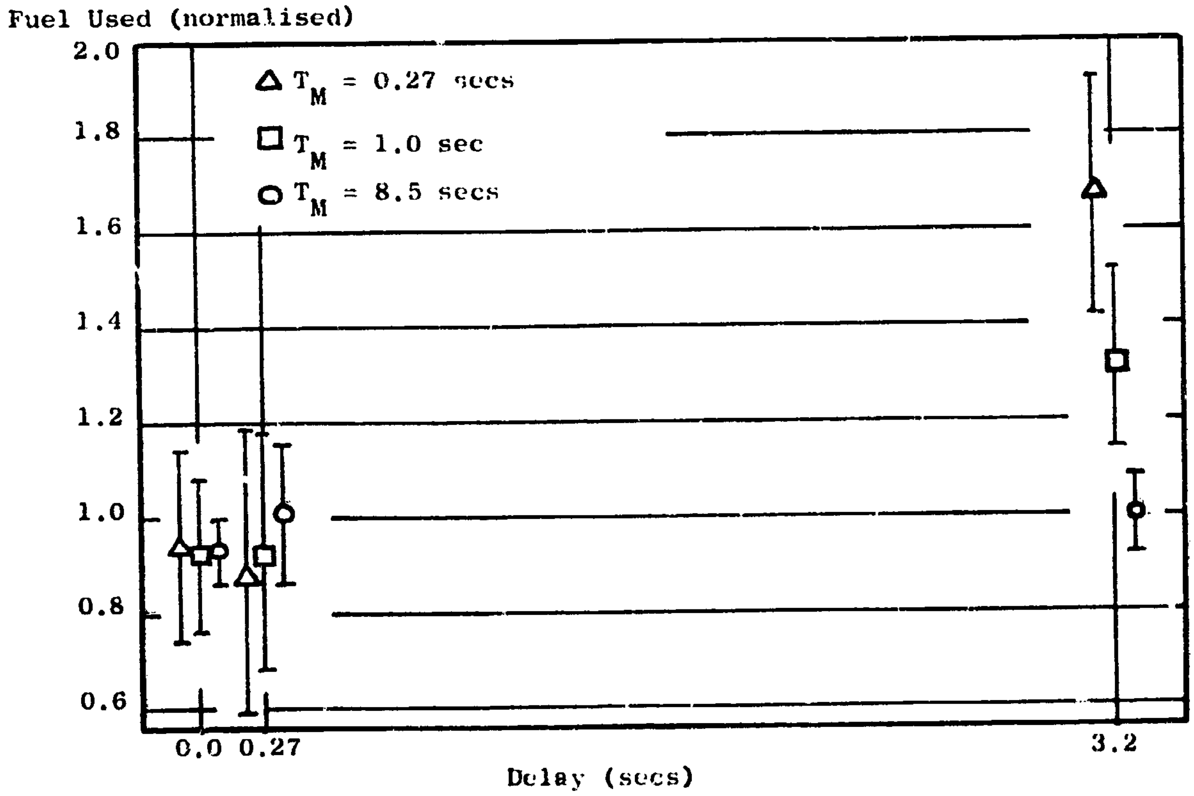


Fig. 21. Effect of Delay and "Delay plus Lag" on Fuel used (Intermittent Vision).

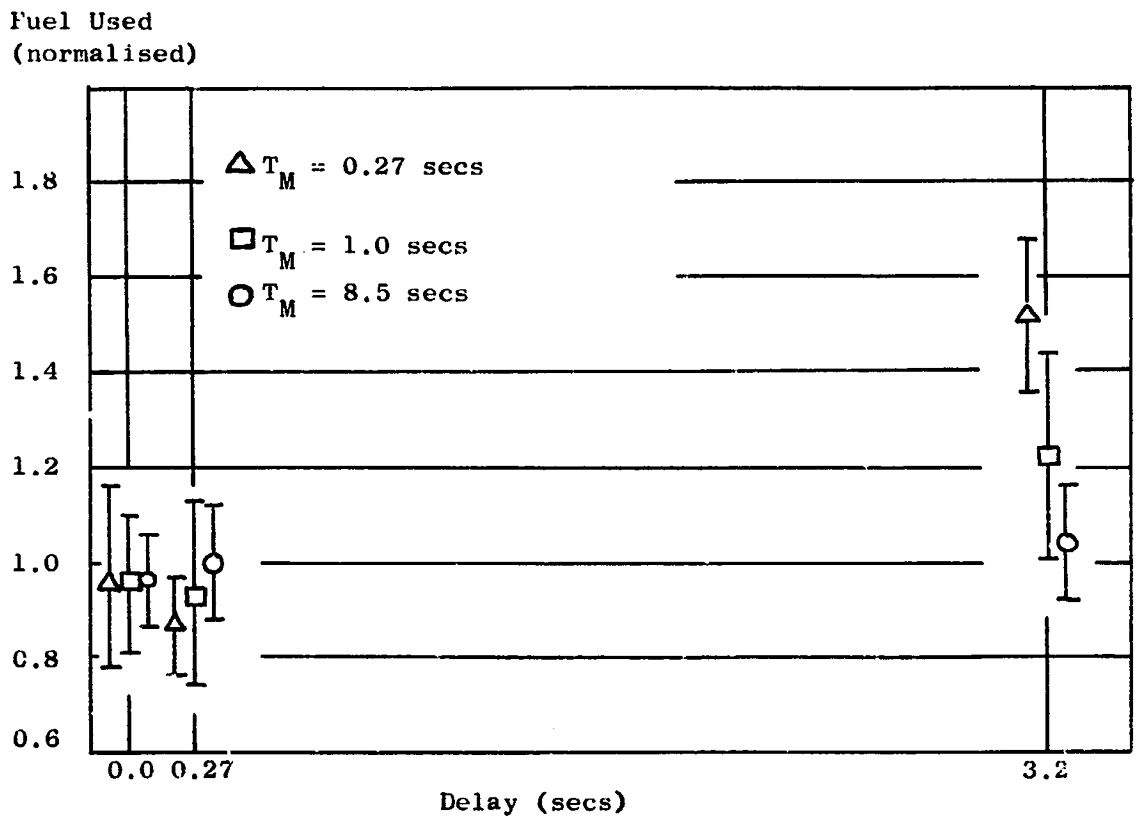


Fig. 22. Effect of Delay and "Delay plus Lag" on Fuel used (Combined Continuous and Intermittent Vision).

With the small dynamic lag, there appears to be a considerable increase in fuel required when the delay was 3.2 seconds. A proportional increase is not found for the large dynamic lag. No explanation is offered for this difference.

5.3 Effect of Lack of Motion Cues on Performance.

During Phase II of the experiment, four experiments were run. Three of these were designed to answer questions raised during Phase I of the experiment, while the fourth was an investigation into the usefulness of some degree of automatic control at the "remote" end of a man-machine system.

One question to be answered was, "How would the operator perform if the intermittent picture presented to him provided no velocity information?" During the intermittent vision experiments of Phase I, subjects had been able to perform accurate control actions during blackout periods by extrapolating from previous visual information. If the operator of a remote machine did not have a television-type display, but instead was seeing a succession of built-up pictures, he might be unable to predict as accurately, and therefore require more commands.

The subject was told to complete the same task as in the previous experiment. There were two display conditions: (1) a view of the continuous display, lasting 0.6 seconds at 4.2 second intervals and otherwise occluded, and (2) a static "snapshot type" picture displayed for 0.6 seconds at 4.2 second intervals and otherwise occluded.

The first subject for this experiment was inexperienced in this task. The results obtained when he performed this task with a 3.2 second delay and 0.27 second time constant are tabulated below.

Display	Time (secs)	Commands	No. of Errors
Static	176 (21.2)	28.3 (3.0)	3
Moving	140 (11.8)	27.3 (1.4)	0

The tabulated figures are the mean and standard deviation for 10 completed runs.

Table 4. Effect of "Static" Vision on Inexperienced Subject's Performance.

The ten "static" runs were completed before starting the "moving" runs. The subject (an experienced observer) stated that he could detect no difference in difficulty between the two conditions (he did not see the experimental printout).

A student's t test shows the difference between the times to be significant at the 0.1% level. The F test for variance ratio shows the difference between the times is significant at the 5% level, and the difference between the commands is significant at the 1% level. These results suggest that this subject did use the motion cues obtained from a TV-like picture, although some learning may have occurred during the experiment.

Further tests were made using two subjects. One (D.B.) was experienced, having been tested during Phase I of the experiment, while the other (K.C.) was inexperienced, having had less than 10 hours training prior to the experiment. Subject D.B. was tested under four conditions: (a) no delay and 8.5 second time constant, and (b) 3.2 seconds delay and 0.27 second time constant; each of these with the same two display conditions, static or moving, lasting 0.6 seconds at 4.2 second intervals. The results are tabulated below.

Time Constant (secs)	Delay (secs)	Vision Time (secs)	Commands	No. of Errors
8.5	0.0	Moving 109 (19.5)*	26.0(4.4)	5
8.5	0.0	Static 115 (13.3)*	26.8(4.9)	1
0.27	3.2	Moving 127 (24.3)**	29.3(5.2)*	7
0.27	3.2	Static 128 (13.9)**	28.0(3.5)*	5

The tabulated figures are the mean and standard deviation for 16 completed runs at each condition.

Table 5. Effect of Static Vision on Experienced Subject's Performance.

**Significant at 2½% level (F test)

*Significant at 10% level (F test)

These figures are the mean and standard deviations for 16 completed runs. The subject completed four runs under one condition and then changed to another condition. The order of the conditions was varied.

The differences between the means for static versus moving display are not significant at the 5% level (t-test). The F-test for the ratio of the variances showed significant differences between the static and moving conditions (see Table 5).

The other subject, K.C., was tested under one condition, no delay and 8.5 second time constant. His results are tabulated below.

Display	No. of Runs	Time (seconds)	Commands	No. of Errors
Moving	10	105.7 (10.6)	25.9 (3.9)	6
Static	12	110.7 (8.4)	25.4 (3.7)	7

Table 6. Effect of Static Vision on Inexperienced Subject's Performance.

The differences between these results are not significant.

From these results, it appears that the subjects did not need the motion cues that could be obtained from a TV-type picture lasting 0.6 seconds. The experienced subject (D.B.) stated that he preferred the static picture because it was less confusing. The inexperienced subject (K.C.) stated that he had no preference for either type of display. It therefore seems possible that the operator of a remote manipulator will experience little difficulty in using a static picture that is updated at intervals.

5.4 Effect of Speed on Operator Performance.

The next experiment attempted to discover what effect, if any, the speed of travel of the fingers had on operator performance. One subject (K.C.) was used, with one condition, no delay, 8.5 second time constant, continuous vision. Four speeds were used: 0.23 inches/second, 0.46 inches/second, 0.92 inches/second, and 1.85 inches/second. The subject had 20 runs at each speed. He did them in groups of 5 runs in varying order. The results of this experiment are tabulated below.

Speed (in/sec)	Time (secs)	Commands	Errors
0.23	100.7 (9.2)	20.0 (3.4)	6
0.46	74.1 (8.6)	21.0 (3.9)	4
0.92	67.6 (12.4)	23.2 (3.3)	4
1.85	62.0 (12.9)	21.3 (3.4)	9

Table 7. Effect of Speed on Subject's Performance.

These results are plotted in Figure 23. The times appear

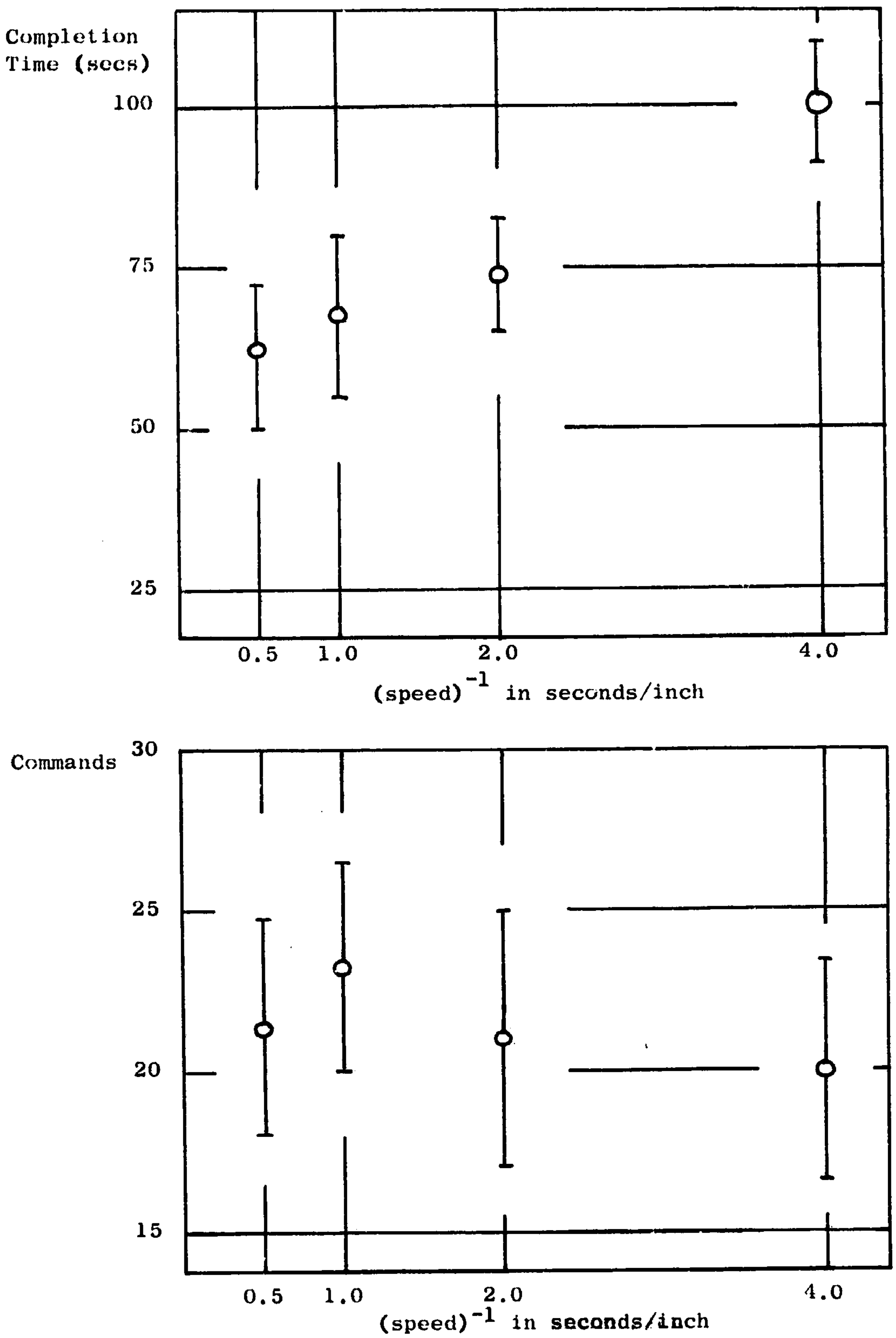


Fig. 23. Effect of Speed on Operator Performance.

to have a linear relation to the inverse of velocity, which might be expected.

The subject considered that the high velocity was more difficult, and this was evident to the experimenter. The number of errors at this speed is not, however, shown to be significant by a chi-squared test.

5.5 Effect of Increased Delay on Performance.

The third experiment investigated longer delays than those used in Phase I of this work. Three subjects were used (one of them the author) with 4 delays, 0.0 seconds, 3.2 seconds, 8.0 seconds, and 12.8 seconds. The time constant was 8.5 seconds. In order to increase the task difficulty, the oversize of the target hole was reduced from 0.21 inches to 0.07 inches, and limitations were placed on the use of the jaws. If the jaws hit the ground, an error was called; and if they gripped the block too hard, an error was called. The task difficulty was increased in order to hold the subjects' interest and to prevent boredom. The constraints, especially with 12.8 seconds delay, require a large amount of effort from the subject. (One subject, D.B., received a round of applause from a visiting audience after completing a run.) The maximum velocity was changed to 0.46 inches/second.

The subjects were given the same instructions as before. They each completed ten experimental runs for each delay.

The results are tabulated below.

Delay secs.	Subject D.B.	Subject K.C.	Subject S.M.
		TIME (secs.)	
0.0	90.4 (6.5)	90.7 (13.2)	72.4 (8.4)
3.2	119.0 (24.4)	126.8 (17.5)	120.4 (25.0)
8.0	173.9 (28.0)	198.4 (25 7)	167.3 (26.5)
12.8	260.0 (38.2)	230.4 (24.8)	287.0 (86.3)
		COMMANDS	
0.0	20.0 (2.9)	22.0 (2.6)	21.5 (3.7)
3.2	27.4 (3.6)	22.3 (2.2)	27.9 (4.0)
8.0	30.4 (3.6)	24.9 (2.0)	30.0 (3.5)
12.8	32.2 (5.4)	24.2 (1.2)	34.9 (8.7)
		ERRORS	
0.0	0	1	3
3.2	3	5	5
8.0	6	10	6
12.8	4	6	5

The tabulated figures are the mean and standard deviation for 10 completed experimental runs for each subject.

The errors are the number of unsuccessful runs made while making 10 completed runs.

Table 8. Effect of Long Delay on Performance.

This data is plotted in Figures 24 and 25. It can be seen that completion time increases approximately linearly with delay. The number of commands also increases with delay for two of the subjects, whereas there is no such evident trend for the third.

One subject, D.B., altered his strategy from using the maximum available velocity for large motions to using about half the available velocity. The other two used the maximum velocity for all large motions.

There is no obvious reasons why the number of commands should increase with delay. This trend was not evident in the first set of experiments, possibly because the task was much easier. The long delays were so long that all moves were "open loop" for the two subjects using maximum velocity, whereas D.B., using half maximum velocity, was able to obtain some visual cues from initial motion. It can be seen that the results for D.B. and S.M. are very similar. It may well be that irritation caused by a long delay added to the long settling time for the large time constant caused a deterioration in their performances.

The number of errors appears to increase to 8.0 seconds, and then decrease with increasing delay, which seems very unlikely. A chi-squared test shows that these differences are significant at the 10% level.

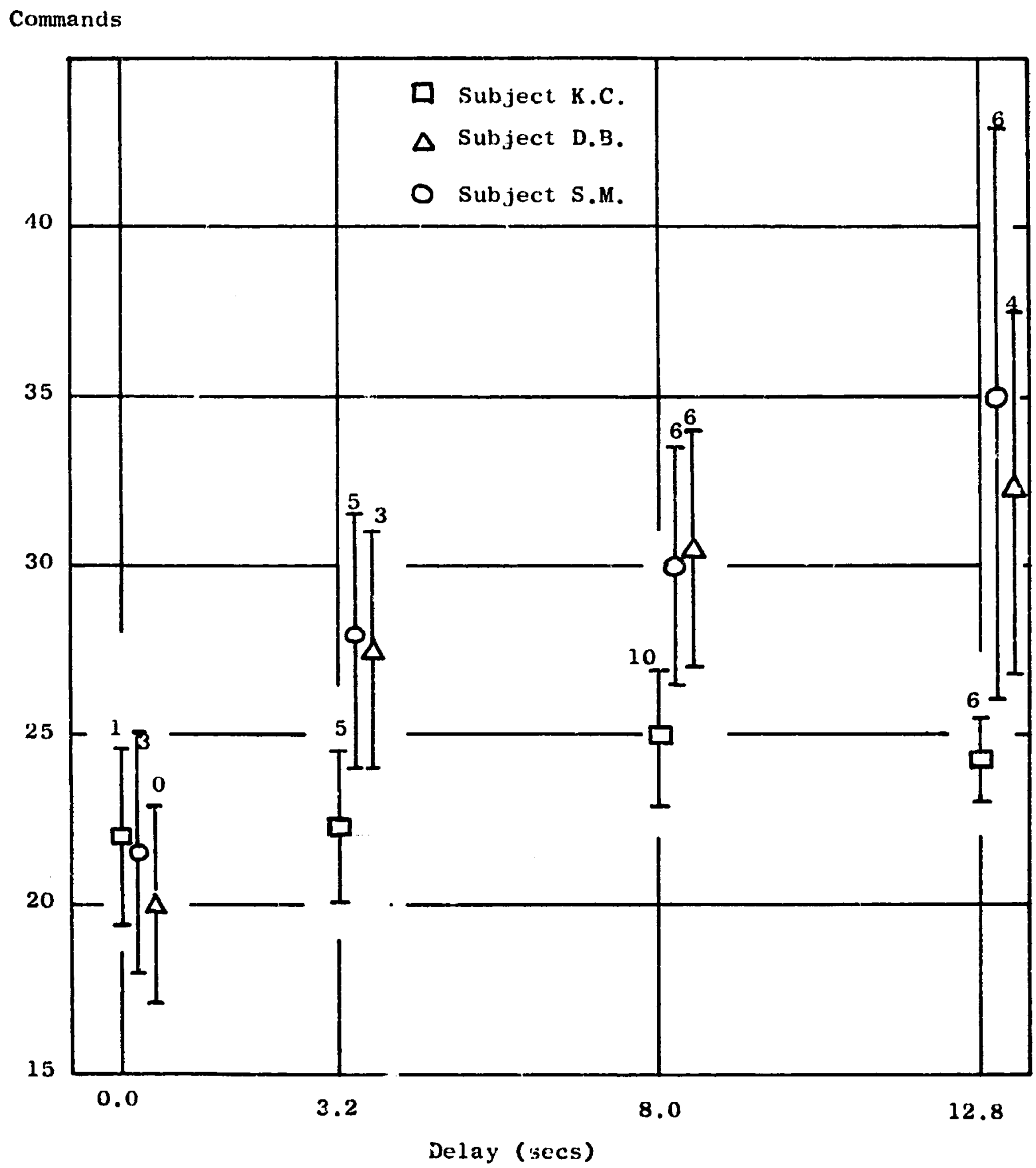


Fig. 24. Effect of Long Delay on Commands Used.

Note: The figures at the top of the standard deviation bars indicate the number of errors made per 10 completed runs.

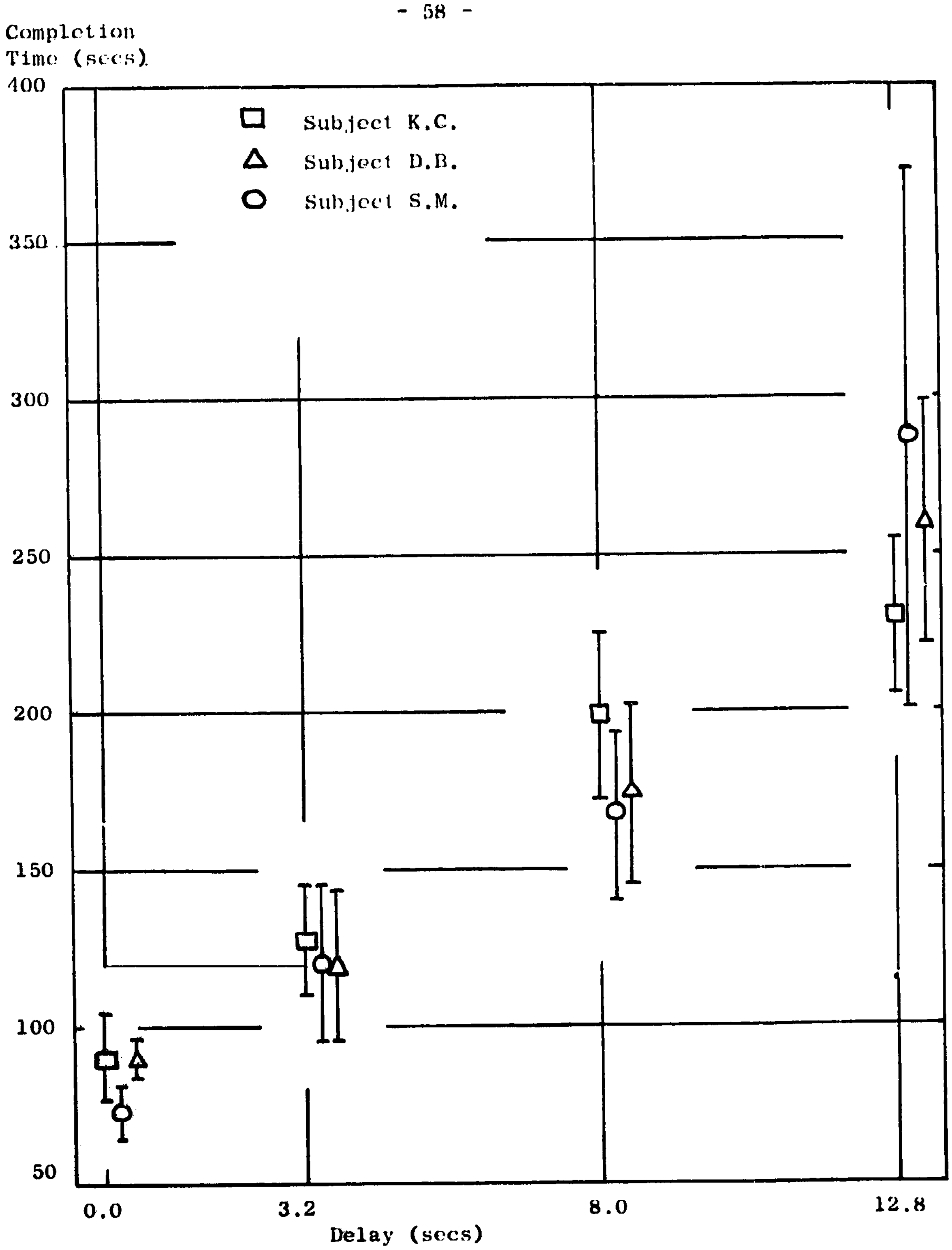


Fig. 25. Effect of Long Delay on Completion Time.

C 2

5.6 Supervisory Control.

The final experiment attempted to discover how effective the special subroutines would be to the operator. Two subjects were used under two conditions, no delay and 12.8 second delay. They were told to use the subroutines to pick up the block, locate it in the hole, and to lower and release it. These subroutines were initiated by the subject by using Switches #12, 13, and 14 on the control panel.

In their initial arrangement, the subroutines were slightly unreliable. This was mainly due to the difficulty of programming an adequate simulation of elastic collisions between bodies. The amount of rebound, and hence the size of the step, varied widely from run to run. A velocity feedback loop inserted in the subroutine solved that problem. The subroutines stopped all motion before proceeding. However, the large time constant present in the system sometimes allowed the block to drift a long way out of position before stopping.

The subjects thought that because the job was now automated, they merely had to press switches, whereas judgment of time durations and estimations of distances and rates was still necessary, although the precision required was far less.

Some preliminary results show that the use of these subroutines seemed to offer little advantage. These results are tabulated here.

Subject	Delay (secs.)	Time (secs.)	Commands
K.C.	12.8	264 (34.5)	24.6 (3.5)
K.C.	0.0	152 (39.5)	22.9 (2.6)
D.B.	0.0	117 (28.3)	19.1 (1.4)

Table 9. Preliminary Results Using Subroutines.

The figures show no great improvement over those obtained without using the automatic subroutines.

The subjects were told to perform the same task as before, using for one set of runs all the subroutines, and for the other set using only two of the three, omitting the first subroutine. As before, there were two delays used. One subject, D.B., also repeated doing the task with no subroutines.

These results are tabulated in Table 10 (page 61).

These results are plotted on Figures 26 and 27.

It can be seen from Table 10 that fewer commands were needed when the subroutines were used. An F test showed significance at the 5% level for subject K.C. for both delays, and significance at the 1% level for the 12.8 second delay for subject D.B. A Students' t-test

Delay secs.	Subroutines Used	No. of Runs	D.B.	No. of Runs	K.C.	
		TIME secs.				
0.0	All	10	139.5 (46.8)	15	128.5 (25.3)	
0.0	2, 3	10	104.2 (18.8)	16	104.5 (16.1)	
0.0	None	10	90.4 (6.5)	10	90.7 (13.2)	*
12.8	All	10	207.3 (28.4)	11	286.1 (55.5)	
12.8	2, 3	13	235.9 (41.8)	11	228.3 (16.4)	
12.8	None	10	210.2 (24.8)	-	- -	
12.8	None	10	260.0 (38.2)	10	230.4 (28.4)	*
		COMMANDS				
0.0	All	10	18.7 (4.1)	15	17.1 (1.5)	
0.0	2, 3	10	18.2 (1.3)	16	18.4 (2.5)	
0.0	None	10	20.0 (2.9)	10	22.0 (2.6)	*
12.8	All	10	20.5 (3.9)	11	16.4 (2.3)	
12.8	2, 3	13	27.1 (5.0)	11	18.4 (1.9)	
12.8	None	10	30.3 (1.6)	-	- -	
12.8	None	10	32.2 (5.4)	10	24.2 (1.2)	*
		ERRORS				
0.0	All	10	1	15	2	
0.0	2, 3	10	0	16	0	
0.0	None	10	0	10	1	*
12.8	All	10	5	11	6	
12.8	2, 3	13	10	11	4	
12.8	None	10	10	-	-	
12.8	None	10	4	10	6	*

*Note: These entries are copied from Table 8, for convenience.

The tabulated figures are the mean and standard deviation for the number of experimental runs listed at each condition.

The errors are the number of unsuccessful runs made while making 10 completed runs.

Table 10. Effect of Subroutine Use on Performance.

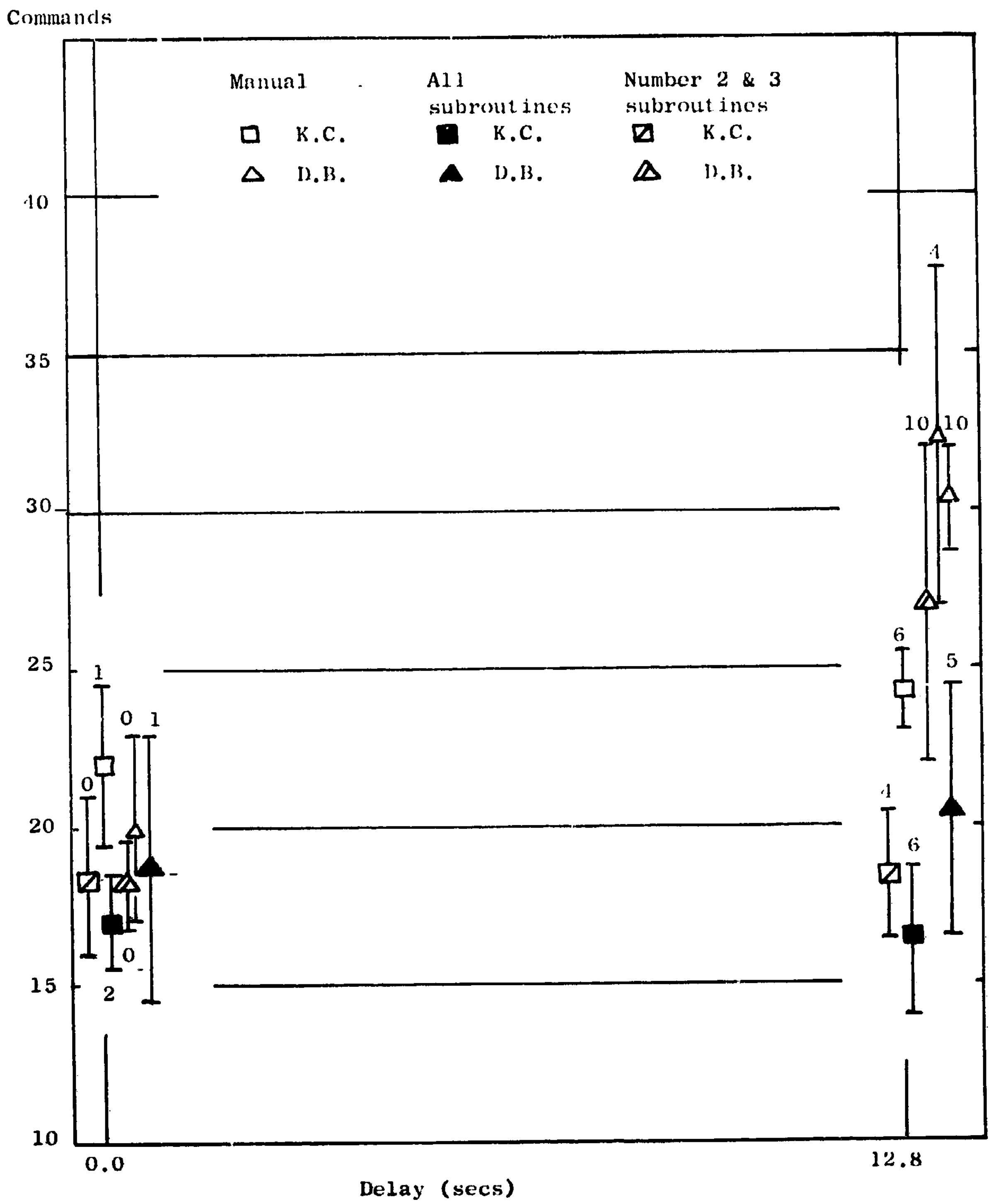


Fig. 26. Effect of Automatic Subroutine Use on Commands Used.

Note: The figures at the end of a standard deviation bar indicate the number of errors made per 10 completed runs.

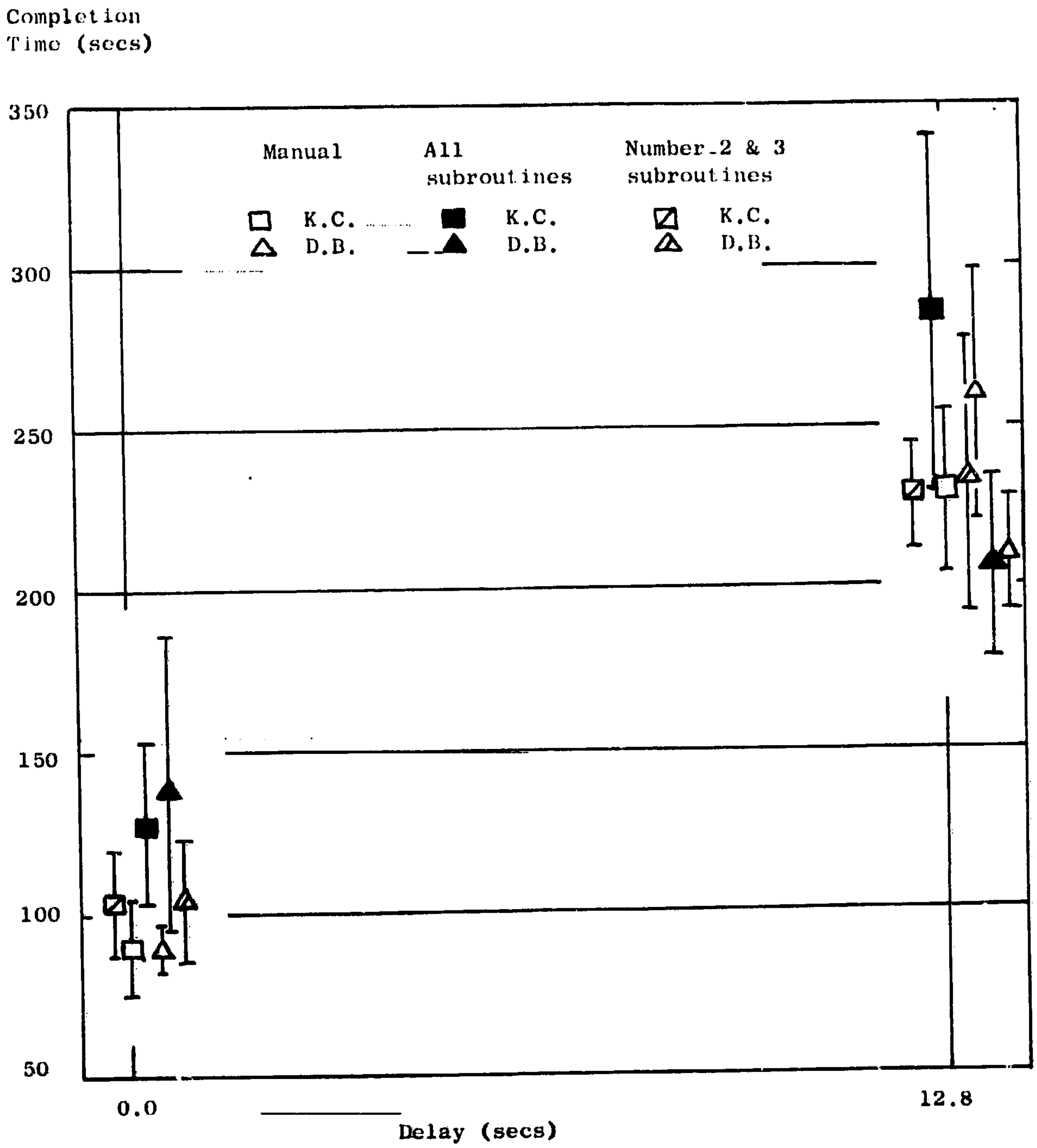


Fig. 27. Effect of Automatic Subroutine Use on Completion Time.

showed differences significant at the 0.1% level for these subjects and delays. The F and t tests show no significant differences between subject D.B.'s earlier and later performances with no delay.

A chi-squared test showed no significant difference between the errors made with or without subroutine use.

Both subjects commented that the subroutines were very useful at the 12.8 second delay. They found the task much easier, and appeared less strained. They did not consider the subroutines useful for the no delay condition. It can be seen from the table that the average completion time with no delay was about 50% more with the subroutines than without them. This added time irritated the subjects.

Subject D.B. objected to performing the task without subroutines with a 12.8 second delay, after he had become used to using the subroutines. His performance, however, was slightly better on this occasion (see Table 10).

With a delay of 12.8 seconds, subject D.B. used notably more commands when he was deprived of the use of subroutine no. 1. He used an average of 6.6 more commands per run. He used 10.1 (1.9)[1] commands to complete the grasp and lift operation when he was allowed to use subroutine no. 1, and 16.2 (2.3)[1] commands when he could not use this subroutine. The difference between these corresponds to

1. Mean and standard deviation.

the over-all difference of 6.6 commands. When not allowed to use any subroutine at all, he used 17.2 (1 2) commands to reach the same point. These figures suggest that this subject's strategy is close to optimal, whereas earlier tests had suggested that the availability of automatic subroutines had made him careless (see Table 9). The error scores are, however, some evidence that subjects are more careless when using the subroutines.

6. CONCLUSIONS

The experiments described in this report support Ferrell's prediction (12) that delayed manipulation is possible using a rate controlled machine, and that operators of such a machine will use a move-and-wait strategy. The time to complete a task increases approximately linearly with delay, at least up to about 12 seconds which was the longest delay used.

At the longer delays, however, the subjects' performance, measured in terms of errors and of commands needed, appeared to deteriorate. The subjects considered a delay of 12 seconds a difficult task.

The provision of relatively low-level feedback and decision making ability to the remote machine made the task much easier for the subject when working with a long delay. The number of commands used with supervisory control was significantly fewer than without it. The supervisory control mode might be expected to reduce the completion times for the long delay condition, by eliminating the numerous waiting periods which occur while the operator waits for confirmation that one move was successful before making the next. The results of the supervisory control experiments do not show this.

The use of a "static" displayed picture instead of a continuous television type display did not seem to impair the subjects' performance. This suggests that a small capacity visual feedback channel might provide enough information for the operator of a remote manipulator through a succession of static "snapshots".

The design and use of this program shows that a small, high-speed computer can be used to provide a moderately interesting experiment. In the real world, however, objects are not always rectilinear, nonslippery and almost indestructible. Manipulators do not have uncoupled motions and infinite room to maneuver. They are not always on a stable platform, and they work in a three-dimensional world. Further development of this work should include some of these difficulties. The simplest of these aspects to incorporate into the existing program would be an "unstable platform".

The automatic subroutines, while adequate for their immediate purpose, might be replaced by a larger set of simpler decision making logical steps. The operator would decide the order of these and then transmit this to the machine.

The advantages of a computer with high-speed input-output facilities and user-compatible software capabilities as compared to laboratory hardware are illustrated by the following example. The addition of a feedback servo loop to one part of the program took 30 minutes. This includes designing the program modification, modifying the program on-line, assembling the modified program, testing the program, and punching out the modified program on paper tape.

REFERENCES

1. Goertz, R. C. Mechanical Master Slave Manipulator. Nucleonics. November 1954.

2. Johnsen, E. G. The Case for Localized Control Loops for Remote Manipulators. Paper presented at I.E.E.E. Human Factors Group Symposium, Boston, May 1965.

3. Yershóv, A.P. One View of Man-Machine Interaction. Paper presented at a Seminar on Automation of Thinking Processes under the auspices of the Scientific Council of Cybernetics of the Academy of Sciences of the Ukrainian Soviet Socialist Republic. Kiev 1963. Translation published in Journal of the Association for Computing Machinery. July 1965.

4. Woodhouse, T. Jacquards and Harnesses. MacMillan and Co., London, 1923.

5. Ernst, H. A. MH1 - A Computer-Operated Mechanical Hand. Sc.D. Thesis (E.E.), Massachusetts Institute of Technology, 1961.

6. Dunne, M. J. The Industrial Robot as a System Component. Paper prepared for the Joint Automatic Control Conference, Rensselaer Polytechnic Institute, Troy, N.Y. Paper XXIV-4. June 1965.

7. Proceedings of the 1964 Seminars on Remotely Operated Special Equipment. United States Atomic Energy Commission reports CONF-640508 and CONF-641120.

8. Kappl, J. J. A Sense of Touch for a Mechanical Hand. S.M. Thesis (Mech. Eng.), Massachusetts Institute of Technology, 1963.

9. Strickler, T. G. Design of an Optical Touch Sensor for a Remote Manipulator. S.M. Thesis (Mech. Eng.), Massachusetts Institute of Technology, 1966.

10. Ferrell, W. R. Remote Manipulation with Transmission Delay. Ph.D. Thesis (Mech. Eng.), Massachusetts Institute of Technology, 1964.

REFERENCES
(Continued)

11. Smith, K. U. Delayed Sensory Feedback and Behavior. W. B. Saunders Co., Philadelphia, 1962.

12. Ferrell. ibid., p. 112.

13. Crawford, B. M. Joy Stick vs. Multiple Levers for Remote Manipulation Control. Wright-Patterson Air Force Base report AMRL-TR-65-76. April 1965.

14. Digital Equipment Corporation. Programmed Data Processor - 1 Manual. Maynard, Mass., 1962.

Appendix I The PDP-1 Computer

(manufactured by Digital Equipment Corp., Maynard, Mass.)

The PDP-1 Computer of the M.I.T. Research Laboratory of Electronics is, when used in the Time-Shared mode, a 4096 word machine, using 18 bit words (14). The user's symbolic program, the text editor, the assembly program, and the on-line debugging system (D.D.T.) are stored on drum memory fields. The user may use these readily and rapidly, using a typewriter keyboard.

The computer will accept data either on tape through a high speed optical reader, or through a typewriter keyboard, or through a bank of switches and knobs. Output facilities include a high speed punch, a typewriter keyboard, and a C.R.T. display, 9¼ inches square, which has 1024 programmable coordinates on each axis.

The programming language uses symbolic language instructions, using 6 bits for memory addressing instructions. The remaining 12 bits of the word provide direct addressing of 4096 registers.

The PDP-1 has a cycle time of 5 microseconds. The central processor has two live registers under program control, the Accumulator (AC) and the In-Out Register (IO). In order to display a point on the C.R.T., its x coordinate is placed in the AC, its y coordinate in the IO, and the instruction "display" (dpy) is given. Each point requires 50 microseconds for display.

The debugging system (DDT) may be used to examine and/or change registers, for example, experimental parameters.

Appendix II — Operation of the Program

This appendix describes the procedure for using the program. This procedure is correct on 11 May 1966; however, changes are made to the PDP-1 programming system from time to time, and any user should confirm from R.L.E. staff that this procedure will work.

The program uses the time-sharing facilities of the PDP-1, but requires "infinite quantum", that is to say, the computer's entire continuous attention, for useful experimental results. Debugging and revision do not require infinite quantum. Infinite quantum is only available at night and during the week end. Specific permission must be obtained from the Research Laboratory of Electronics to use the PDP-1 during these times.

The instructions below cover the following activities: (a) Loading the Program; and (b) Using the Program.

(a) Loading the Program.

Switch on a console. (The typewriter should type "hello" back.)

Type "edit", then "carriage return".

Push down Sense Switches on your console.

Load the tape into the reader with the sprocket holes nearer the machine. Arrange the leader in the left hand box.

Switch on the reader.

Type "r", then "carriage return".

Watch the tape and riffle it.

When it is all read in, turn off the reader.

Provided that it was read in correctly, proceed to assemble.

Type "j", then "carriage return".

Type "N".

Type "s".

The computer should then type "JAWS - pass1".

Wait for a completion pulse.

Type "s".

The computer should then type "JAWS - pass2".

Wait for a completion pulse.

Type "s".

Type "b", then "carriage return".

The computer should then type "dismissed".

Type "ddt", then "carriage return".

Type "2T".

Wait.

After an interval, the computer will type something like

<op 7> .

Type "1U".

The program is now loaded.

(b) Using the Program.

Push up sense switches 2, 3, 6.

Type "4G", then "carriage return".

The computer will then type several lines and punch some tape, then type "Punching completed", then wait for either the experimenter or the subject to do something. If the subject were to press Switch #3 on his panel, an experimental run would start. At this point, however, the experimenter wishes to change some parameters. In order to do this, first touch the "call" button on the console. The current location and its contents will be typed out. Ignore these.

Suppose, for example, that the experimenter wants to change the contents of register "mss" to 3.

He types "mss/".

The computer types "bg2+17" (say).

He types "=".

The computer types "77".

He types "3", then "carriage return".

The complete line of typescript looks like this:

```
mss/     bg2+17  =  77     3
```

In order to change the subject's initials, the procedure is

Type sjt/

The computer types "214522" say,

Type "~".

The computer types "sgm".

Type (say) "jqp"", then "carriage return".

The typescript line looks like

sjt/ 214522 sgm jqp" ___

sjt is the only register that has to be altered in this way.

The experimental parameters that can be varied in this way include:

mss,	the mass or time constant
spd,	the gain between knob setting and steady state velocity
dly,	the transmission delay between the knobs and the system
tol,	the horizontal tolerance available for fitting the block into the slot
eps,	the vertical tolerance on overlap
tm 1,	these are time settings
tm 2,	to determine the nature
tm 4,	of the display presented to the operator
co 1,	the cost of time
co 2,	the cost of display
co 3,	the cost of fuel
co 4,	the cost of commands
gty,	the gravity field

The parameters mss and spd are used as shifting instructions in the program. They may have the following values:

mss, 2, 3, 7, 17, 37, 77, 177, 377

(do not use mss = 1)

spd, 5001, 5003, 5007, ... 5377

(The 5000 is part of the shift instruction.)

The parameters tol and ops are simple linear dimensions.

The parameters dly, tm 1, tm 2, and tm 4 depend on the program cycling speed, which is 17 octal cycles per second. Therefore, if register dly contains 17, there will be a transmission delay of 1 second. The use of registers tm 21, tm 2, and tm 4 is better understood if the display control and coordinate transfer sections of the program are understood. The constraints on the contents of these registers are:

$$0 \leq \text{dly} < 400$$

$$1 < \text{tm } 1 < \text{tm } 2$$

Examples: tm 1 = 100, tm 2 = 101, tm 4 = 0 gives a continuous display of the task.

tm 1 = 16, tm 2 = 17, tm 4 = 170 gives an interrupted display of the task, with a cycle time of 9 seconds with the display on for one second.

tm 1 = 2, tm 2 = 17, tm 4 = 36 gives a snapshot-type display which is on for 1 second every 3 seconds.

The registers co 1, co 2, co 3, and co 4, may be used to contain "costs" of various "expenditures" incurred while performing the task. Suggested values are: co 1 = 10, co 2 = 10, co 3 = 1000, co 4 = 10. These are all linear multipliers.

The registers run, dte, mth, yr are useful for bookkeeping. The register run is indexed after each run.

After setting the content of these registers, the subject may start a run by pressing Button #3. The computer will type "knobs" if any switch except #9, 16, 17 is 'on', or if knobs #0, 3 are not set at the zero point.

The subject operates the jaws through these controls (see Fig. 4):

L. H. Knob	x velocity
R. H. Knob	y velocity
#9 Switch	lets the computer see the velocity knobs
#8 Switch	opens or closes the jaws
#17 Switch	leave down
#16 Switch	down for open, up for close jaws
#5 Switch	finishes the run
#2 Switch	will provide a brief display
#11 Switch	provides a continuous display
#12 Switch	starts subroutine #1
#13 Switch	starts subroutine #2
#14 Switch	starts subroutine #3

At the finish of a run, the program will produce typeout and punchout under control of sense switches #2, 3, 6. With all of these up, a complete set of output is produced. Pushing #3 down shortens the typeout. Pushing #2 down removes the punched output. Pushing #6 down eliminates all output. The punched tape may be read and typed out by the flexowriters used for off-line tape generation.

This typeout is a coded and timed list of the subject's "commands" during the run. The first two digits identify the command, the last four tell the time when the command was given.

The command identifying codes are:

01 start subroutine #3

02 start subroutine #2

04 start subroutine #1

(Note: these three may be "anded" together.)

05 no subroutine on

21 increase x velocity

22 decrease x velocity

31 decrease y velocity

32 increase y velocity

41 close jaws

43 open jaws

45 stop opening or closing jaws

51 demand vision

53 demand vision

The last four digits are generated by the program cycle counter, shifted so that each increment represents approximately one half second.

An example of the typeout is given in Fig. 13. An explanation of the coded information follows:

Type 60D30

First digit 6 shows that the register mss contains 6 bits, i.e., mss = 77;

Second digit 0 indicates continuous vision;

Second digit 1 indicates intermittent vision;

Third character D indicates a non-zero delay;

The remainder is half the number contained in the register dly.

The instructions given above are sufficient to load and run the program, provided that no fault occurs. It is unlikely, however, that no error will ever occur during extended use of this program. A knowledge of the basic time-sharing system may be obtained from the note, "An Introduction to the PDP-1 Time-Sharing System Programs. PDP-29-1. E. E. Dept. M.I.T." which is written for a beginning user.

Note: The user will become familiar with the time that the time-sharing system takes to carry out the activities ordered by "N", "S", "1U", "2T", etc. If, on any occasion, these times vary widely, something has gone wrong. Particular note should be taken of the time taken to do "2T", after modifying the symbolic text. If this takes about 3 seconds, it may be necessary to return to your text, punch out, and start again from the beginning.