ESD-TR-67-289
ESTI FILE COPY

ESD RECORD COPY
RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

ESD-TR-67-289

ESD ACCESSION LIST
ESTI Call No. 66419
Copy No. 1 of 2 cys.

(Final Report)

# INVESTIGATIONS IN COMPUTER-AIDED INSTRUCTION AND COMPUTER-AIDED CONTROLS

April 1967

R. C. Rosenberg
S. G. McCandlish
T. B. Sheridan

DECISION SCIENCES LABORATORY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 7682
Task: 768204

This document has been approved for public release and sale; its distribution is unlimited.

(Prepared under Contract No. AF 19(628)-3317 by the Engineering Projects Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology.)

AD0655374

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

## DDC AVAILABILITY NOTICES

Qualified requesters may obtain copies from Defense Documentation Center (DDC). Orders will be expedited if placed through the librarian or other person designated to request documents from DDC.

Copies available at Office of Technical Services, Department of Commerce.

ESD-TR-289 DSR 9960-4

Final Report

# Investigations in Computer-Aided Instruction and Computer-Aided Controls

R. C. Rosenberg
S. G. McCandlish
T. B. Sheridan

Decision Sciences Laboratory
Electronic Systems Division
Air Force Systems Command
United States Air Force
L. G. Hanscom Field
Bedford, Massachusetts

Project 7682 Task 768204

This document has been approved for public release and sale; its distribution is unlimited.

(Prepared under Contract No. AF 19(628)-3317 by the Engineering Projects Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology)

## FOREWORD

One of the research goals of the Decision Sciences Laboratory is the development of design principles for automated training subsystems which could be built into future Information Systems. Such subsystems would provide Information Systems with the capability of automatically training their own operators and users. The need for such on-the-job training capability has already become apparent. To be able to design such a capability requires first the solution of many conceptual and experimental problems.

Task 768204, Automated Training for Information Systems, under Project 7682, Man-Computer Information Processing, was established to formulate and answer some of these questions. This report is one in a series supporting Task 768204. Dr. Sylvia R. Mayer served as Air Force Task Scientist and Contract Monitor.

This technical report has been reviewed and is approved.

JAMES S. DUVA
Technical Director
Decision Sciences Laboratory

CHARLES A. LAUSTRUP
Colonel, USAF
Director, Decision Sciences Laboratory

## ABSTRACT

This report is a summary of two research projects performed under USAF 19(628)-3317 in the area of computer-aided instruction and man-computer interaction. In 1965 Rosenberg completed a study entitled Computer Aided Teaching of Dynamic System Behavior published as ESD TR-66-260. This study demonstrated that within the delimited area of formal engineering theory a computer simulated laboratory could be built in which freshman students could pose problems, observe displays of machine responses, and thereby learn formal discipline with only minor interaction with a human teacher. An outline of Rosenberg's experiment and conclusions is given in Section I, along with some more general "Observations on the Use of Computers in Instruction," dealing with prospects for computerized tracking monitors. In June 1966 McCandlish completed a study entitled A Computer Simulation Experiment of Supervisory Control of Remote Manipulation, to be published concurrently with the present report as a separate ESD document. McCandlish summarizes how, for a formally well-defined task like grasping a block with a pair of jaws, removing the block from a hole and placing it in a second hole, the human can perform the task through the computer, but only with certain difficulties. He investigated such display rate, time delay between human responses and knowledge of results, and nature of command statements. An outline of McCandlish's experiment and conclusions is given in Section II, followed by "Observations on the Relation Between Computer-Aided-Instruction and Computer-Aided-Control," the latter being where the human serves as a supervisor and subgoal setter of a lower level semi-automatic system which interacts directly with the environment to accomplish a task.

## LIST OF FIGURES

1. A hypothetical example of computer-assisted drill

2. An example of a dialogue interaction

3. An example of a tutorial interaction

4. Examples of problems suitable for student-programming systems

5. Comparison of systems for computer-aided instruction and computer-aided control

## SECTION I A

### SUMMARY AND CONCLUSIONS OF A STUDY ON COMPUTER-AIDED TEACHING OF DYNAMIC SYSTEMS BEHAVIOR* (R.C. Rosenberg)

A Dynamic Systems Laboratory, which is based on a time-shared digital computer, was built for the purpose of teaching dynamic system behavior. The Laboratory simulates the behavior of linear lumped-parameter dynamic systems, which are specified by the student in bond graph notation. All of the dynamic response variables (e.g., voltages, displacements, powers) are available in the form of tables or plots, and high speed response plots are generated on an oscilloscope display set up by a special analog computer. Thus a close temporal association between the description of a dynamic system and a display of its response characteristics is available to the student.

The principal application of the Dynamic Systems Laboratory was in an experiment on the teaching of introductory dynamic systems behavior, in which the Laboratory served as the basis of a self-instructional system. The Dynamic Systems Laboratory was used to develop a general conceptual framework for explaining the behavior of classes of dynamic systems based on qualitative investigation and quantitative formulation of experimental data by the students. This is in contrast to the usual lecture and textbook style of straightforward exposition for concept presentation. Such an application of the Dynamic Systems Laboratory was found to be inefficient for all the subjects, and ineffective for one. For studying the behavior of systems in detail the pattern of thoughtful prediction followed by experimental verification proved effective and motivating to the subjects. However, when guesses, rather than reasoned predictions, were made the verification procedure became quite inefficient. A test facility, in which the subjects tested and synthesized unknown (or black box) systems, was found to be valuable for several purposes, which included the measurement of achievement, the increase of understanding of system

behavior by the subjects, and the motivation of the subjects.

The Dynamic Systems Laboratory was used as a classroom teaching aid, and assisted the lecturer in conducting a flexible design and control study of a large, complex dynamic system. In addition there was an increase in student participation in the class discussions.

The primary role of the Dynamic Systems Laboratory in a fully automated instructional system should be to provide a responsive environment for applications of the basic concepts, progressing from analysis to prediction and verification to testing and synthesis. A secondary role, that of teaching students sound experimental technique, seems promising, but will require further investigation.

SECTION I B

SOME OBSERVATIONS ON THE USE OF COMPUTERS IN INSTRUCTION (by R.C. Rosenberg)

## 1. Introduction

The predominant form of computer-assisted instruction to date has been the pre-programmed approach stemming from Skinner's programmed instruction concepts (1)*. The type of student-computer interaction has ranged in complexity from simple drill (2) to the rather sophisticated "Mystery Problem" dialogue (3). The major effort has been given to the development and use of tutorial systems (4,5), which present material in a style close to that used in non-computer-based programmed instruction. A brief example of each type of interaction is presented in Figure 1, 2, and 3, respectively.

These efforts are called "pre-programmed" because the teacher must specify every allowable response which can occur. In the Skinnerian view this is desirable, because the teacher is forced to anticipate in great detail the possible responses a student might make, and to plan the requisite feedback action for each response. The logical extension of such an approach to pre-planning is represented by Smallwood's decision structure concept (6), which requires a meticulous analysis of the structure of the material to be presented.

There is another, somewhat different, form of computer-assisted instruction which typically has arisen in the following way. A problem-oriented language is developed to permit a person (usually a researcher) to investigate in convenient fashion a class of problems. Examples from engineering are Stress and Enport (8). It usually occurs to the author or experienced user of these programs that the system should be a valuable tool for teaching about the problem area. However, it is not clear how the inexperienced student can or should be taught by such a system,

*Numbers in parentheses refer to the bibliography.

DRILL ON LEAST COMMON MULTIPLES

| | | | | | | |
|---|---|---|---|---|---|---|
| LCM | OF | 100* | 30* | 5* | IS.... | 300 |
| LCM | OF | 12* | 6* | 5 | IS.... | 60 |
| LCM | OF | 4* | 8* | 6 | IS.... | 48 |
| | SORRY, TRY AGAIN | | | | .... | 24 |
| | RIGHT | | | | | |
| LCM | OF | 30* | 7* | 21 | IS.... | 210 |
| LCM | OF | 1* | 19* | 3 | IS.... | 57 |

END OF LCM DRILL 3.

NO. PROBS. 5
NO. CORRECT 4
PER. CORRECT 80

(Note: Student's responses are underlined)

Figure 1. A Hypothetical Example of Computer-Assisted Drill.

(The following is a partial transcript from a program intended to assist medical trainees in learning to make diagnoses.)

.
.
.
.

TEMPERATURE?

103.

BLOOD PRESSURE?

150/84

PULSE?

100 a minute. Occasional extra systole

ABDOMINAL EXAM?

The EENT exam comes first!

EENT EXAM?

Essentially negative.

.
.
.
.

ABDOMINAL EXAM?

Diffusely tender, some splinting noticed in lower right quadrant and occasionally radiating to right flank area.

THE GRIPPE.

There's insufficient information for any conclusive diagnosis at this time. You're not being careful!

.
.
.
.

Figure 2. An Example of a Dialogue Interaction.

(Source: Feurzeig, W. et al: Somputer-Aided Teaching in Medical Diagnosis, J. Med. Ed., V. 39, N. 8, (Aug. 1964), p. 746-754.

The equation $y = mx + b$ describes a straight line. What kind of line does $y = 2x + 4$ describe?

(A) straight (B) curved (C) don't know <u>A</u>

-RIGHT.

What kind of line does $y = 3x$ describe?

(A) straight (B) curved) (C) don't know <u>A</u>

-RIGHT.

"m" is called ths lope of the line.
What is the slope of the line $y = 3$?

(A) 3 (B) 1 (C) 0 (D) don't know <u>A</u>

-The slope is 0. Substitute "0" for "m" in the equation $y = mx + 3$ to see this.
What is the slope of the line $y = -x-8$?

(A) -8 (B) 8 (C) -1 (D) 1 (E) don't know <u>C</u>

-GOOD.

(Note: Student's responses are underlined.)

Figure 3. A Hypothetical Example of a Tutorial Interaction.

The predominant characteristic of these "student programming" systems is that the student can construct a large variety of responses, within the confines of the language. These responses produce specific information for the student; the system is controlled by him.

What can be done to help a student learn effectively from a student programmed system? For what types of educational objectives are student-programmed systems most appropriate? In the next section these questions are discussed.

## 2. The Use of Student-Programmed Systems

In what educational context should a student-programming system be placed? It's principal characteristic of being able to respond to a wide range of student conjectures and formulations suggests that emphasis be placed on having the student attempt problems of formulation. To be more specific, consider the simple, rather general purpose programming system named Telcomp (9). The most important parts of the language for our purposes are:

| | | | |
|---|---|---|---|
| DEMAND | variable list | ; | input |
| SET | variable = expression | ; | computation |
| TYPE | expressions or variable list | ; | output |
| TO STEP n IF Boolean expression | | ; | conditional transfer |

A student equipped with just these four commands may be posed a problem to develop an algorithm to

--find the lowest common multiple of A, B, and C; or
--find the largest number in a list; or
--find the real roots of an $n^{th}$ order polynomial; and so on.

It is evident that the task of generating material to enable a pre-programmed system to pose the same problems is enormous. Most likely no one will be tempted to try more than a demonstration problem. The point is that a system like Telcomp makes available to a student the ability to solve a wide class

of problems.

In general one would expect to make use of a student-programming system of instruction by presenting a series of problems in construction or formulation (perhaps annotated with answers or representative data on performance). Rosenberg (8; pages 84-88) attempted to classify use of a student-programmed system for dynamics into four categories:

(1) qualitative observation of phenomena (e.g. observe the voltage on resistor R1);

(2) quantitative study of phenomena (e.g. what kind of curve does the voltage on R1 follow?);

(3) prediction and verification (e.g. for what R1 value will the steady-state voltage be 100?);

(4) testing (similar to prediction, but without the student verification step).

There were insufficient data for each category of use to permit any firm conclusions to be drawn. However, testing proved to be a uniformly successful application, while it seemed that suggesting qualitative investigation of phenomena without good feedback is a risky pedagogical strategy--it permitted misconceptions to become somewhat ingrained.

## 3. Monitoring in Student-Programmed Systems

Since by design the interactions between the student and the computer are quite complex in a student-programming system, it is not feasible to construct a simple teaching model of the programmed instruction form and apply it directly. It is apparent that one of the key contributions of any computer-assisted instructional system is the feedback which can be provided to the student. The monitoring question may be stated as--how can appropriate feedback be provided to a student who is using a student-programming system? Figure 4 poses three different problems, each relevant to a

a. Develop a procedure to do the following: "given three whole numbers--a,b,c -- find the least common multiple."

b. Solve the following algebra word problem (the computer is available to do the algebra): "On an examination of n questions a student answers correctly 15 of the first 20. Of the remaining questions he answers one-third correctly. Al the questions have the same credit. If the student's mark is 50%, how many different values of n can there be?"

(A) 4 (B) 3 (C) 2 (D) 1 (E) the problem cannot be solved

c. Describe the motion of the mass in the following problem:

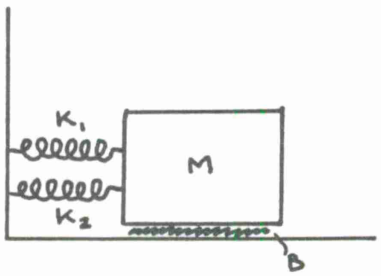


Figure 4. Examples of Problems Suitable for Student-Programming Systems.

distinct type of system.

Let us consider some monitoring requirements for the algebra word problem (see Fig. 4b) (taken from Ref. 10; page 65). One way in which to analyze the problem is to define a set of useful, or potentially useful variables, and attempt to construct a sufficient set of relations among them. That this may be done algorithmically has been shown by Bobrow, who developed the STUDENT program for solving certain types of algebra word problems (11). If an actual student is permitted to define any variables he wishes, and to construct a set of relations among the variables, what are some ways in which an instructional monitor could assist him?

It could comment upon his variable set; there may be redundant, irrelevant, and/or incorrect variables named;

it could examine and discuss the realtion set, which may contain incorrect, incompatible and/or irrelevant relations;

it could, rather than present direct feedback or evaluation, suggest a related problem to be considered by the student;

it could merely show the student how the STUDENT program solved the program.

To sum up and generalize some of the contributions of a monitor might make in instructional use of a student programming system, we find that--

a student may be asked to formulate, as well as solve, problems;

he may choose his own notation and variables;

he may decide what data are required for an adequate (i.e. solvable formulation;

he may establish the formulation procedures in his own style; and he may arrive at the results from his formulations either unaided, or assisted by the computer's capabilities.

In all of these activities a good monitor would provide guidance to the student while restricting the student's responses only as much as called

for by the instructional plan.

## 4. Strategies for Realizing Some Monitoring Capabilities

In order to provide some of the monitoring capabilities just indicated, five strategies are discussed briefly here.

### 4.1 A pre-programmed approach

It is possible to achieve some of the monitoring capabilities discussed earlier if a sufficient number of possible responses of the student are anticipated. It is not likely that the student could select his own notation, or very much of his own procedure, since the tree of possibilities would grow very large very rapidly. The tendency would most likely be to lead the student in a pre-determined "correct" direction for formulation and solution, due to practical considerations. Such an approach to monitoring represents an attempt to extend the currect techniques in the application of programmed instruction to the use of student programming systems.

### 4.2 A non-specific monitor

If a set of non-specific comments on student activity can be found, and if they can be properly keyed to the student's sequential development of the problem (much in the fashion of a non-directive psychiatrist), then a student may be helped to an awareness of relevant considerations in formulations, although indirectly. Such a set of comments would be intended to create in the student the ability to carry out the monitoring function himself, by helping him consider his own procedure and seek relevant feedback.

### 4.3 An "intelligent" monitor

If a meta-structure for modelling the formulation procedure can be found (thus far an extremely difficult challenge for any class of non-trivial problems), then it may be possible to construct a monitor which can assess

the student's current state in the structure. Such a monitor may then guide the student through the formulation procedure in a responsive, but appropriate, way. In essence, the monitor "understands" where the student is currently in the problem, where he needs to get to, and how to guide him there.

### 4.4 A "here's how I do it" monitor

If a computer can be programmed to solve the same formulation problems posed to the student, and if its procedures can be meaningfully interpreted in terms of human capabilities, then a student may be able to learn by interrogating the program as to its procedures (i.e. look over its shoulder). This strategy is a possibility in algebra word problems, for example, since a program to solve such problems does exist (see 11).

### 4.5 A group approach

It may be that two or three students of approximately the same background could interact fruitfully in relation to a student programming system. Collectively they may verbalize the procedures for attempting formulation, construct hypotheses, try to verify them, and consider the implications of errors. Some evidence for the positive effects of groups is to be found in the data on simulated business and decision-making games. Clearly, there is the danger of an unhealthly interaction in which conjectures are suppressed for fear of error and a dominant figure sets the pace and pattern. A group approach has the potential secondary advantage of being relatively more economical in student/terminal ratio; it is not established whether the student/computer time would be improved. Allied with this technique is the possibility of gaining insight into student learning processes by taperecording group interactions.

## 5. Summary

The purpose of this report has been to point out the existence of a class of computer-assisted instructional systems called student-programming systems. These are distinguished by having general problem-oriented language , and permitting the student great flexibility in setting up his

own problems. A suggestion of how such systems might be used with respect to some subject matter was made. The problem of monitoring a student to provide him with helpful feedback in formulation problems was considered, and a set of possible strategies for realizing some monitoring functions was described.

## SECTION II A

## SUMMARY AND CONCLUSIONS OF A COMPUTER SIMULATION EXPERIMENT OF SUPERVISORY CONTROL OF REMOTE MANIPULATION (S.G. McCandlish)

The long term aim of this work[12] is the modeling of the process by which the human commands and controls a real-time information system containing automatic subroutines which may be used to accomplish portions of a task. Remote manipulation is believed to have all the typical attributes of such a system yet be simple enough to be amenable to laboratory investigation.

The replacement of men by remotely operated manipulators is desirable in harzardous 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.

Results indicate that delayed manipulation is possible using a continuous 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. Note we do not mean delay in the evaluation of response, as reenforcement in a learning experiment. It is a delay of any knowledge of results whatsoever.

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 time 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 operator still waited, but with an obviously decreased perceptual burden.

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". Anecdotal evidence suggested that the operators would prefer occasional "snapshots" during the execution of a subroutine to assure themselves that the correct action was being taken and that time was not being wasted.

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

The advantages of computer with high-speed input-output facilities

and user-compatible software capabilities as compared to laboratory hardware are illustrated by the following example. The addtion of a feedback servo loop to one part of the program took 30 minutes. This included 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.

## SECTION II B

### OBSERVATION ON THE RELATION BETWEEN COMPUTER-AIDED INSTRUCTION AND COMPUTER-AIDED CONTROL (by T.B. Sheridan)

The experiment of McCandlish seeks to understand how a human operator controls a semi-automatic tool which in turn controls a passive manipulated object. The purpose of the system is to achieve a specified change in the state z of the manipulated object. This is accomplished by the human operator through an ordered sequence of subgoal statements interpretable by the particular automatic tool available. The latter in turn achieves each subgoal through its own mechanism (sensors, data processing, memory store, effectors). This situation is diagrammed in Figure 5a.

Two types of feedback variables are represented. The first, z', is the actual manipulated object state z as measured by an "immediate performance evaluator" with reference to the particular subgoal of interest and the capabilities of the automatic tool for achieving this goal. The second feedback variable, z'', is the manipulated object state as measured by a "long term performance evaluator" with reference with the long term goal x. This paradigm presumes to apply to a variety of situations where humans sequentially command automatic devices to achieve subgoals.

The kind of computer-aided simualtion experiment discussed by Rosenberg (Section I A) can be represented by a similar diagram where the automatic tool is an information system, the manipulated object is a passive physical process and both are simulated on a computer. However, when Rosenberg discusses the teaching monitor (Section I B) the roles of the human and mechanical information system are reversed, (Fig. 5b). The subgoal setting is taken over by the mechanical device in the latter case, and the subgoals are to be set so that they can be achieved by the particular human

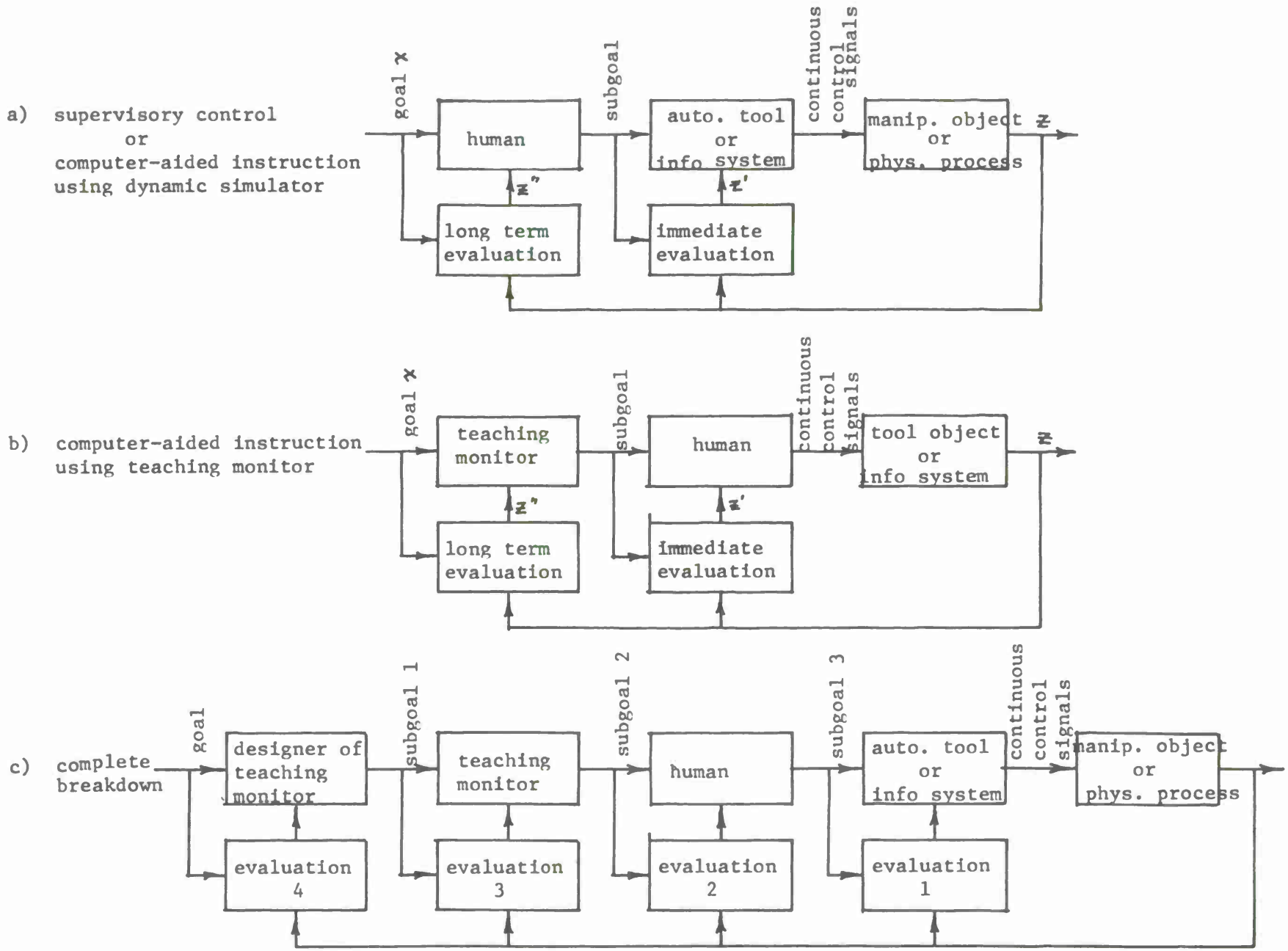


Figure 5 Comparison of systems for computer-aided instruction and computer-aided control.

operating in the system and the constraints on him. As before, the purpose of the latter system, for any one task, is to achieve a specified change in state of the real or simulated physical process (which is really the only operational measure of whether the human has "learned").

The problem of what is learning and what is learned is elusive. Obviously, for a repeat of precisely the same task, the automatic manipulator could simply have stored the sequence of subgoals presented and later repeated the whole movement pattern with no further help from the human supervisor. Likewise the human, should he be training for a precise repeat of some response pattern, could attempt to remember the subgoal sequence presented him on the first round by the teaching monitor.

The real problem of the teaching monitor is to choose efficiently that sequence of subgoals which will elicit the response patterns from the human to accomplish the task. Usually the more efficient (the shorter the subgoal list) the better the system will be. Learning, whatever it is, will follow. This is true for learning a single response pattern or for training to accomplish a complex task calling for different responses contingent upon different stimuli.

The designer of the teaching monitor can be viewed as a subgoal setter for the teaching monitor, and so on, creating a heirarchy (Fig. 5c) alternating between human and mechanical devices.

Specification of the higher level function is obviously more difficult than specification of the lower levels. For this reason design of simulators for teaching purposes (really a device which forces the human subjects <u>into self-conscious sub-goal setting roles above itself</u> appears an easier avenue for teaching technology to proceed than the design of a teaching monitor which sets subgoals for humans below itself.

Engineering the more primitive device beneath the human is simpler than that above him. This will still accomplish the aim of forcing the human to be conscious of his logical coding technique (subgoal setting) and thereby remember the structure of what he does and what results ensue.

The "tightness of connectivity", or its converse, the degree of authority delegated from higher level to lower level, be it from teaching monitor to human learner or from human supervisor to automatic control device, is of interest. A very tightly connected system is characterized by time constants of operations in inner or lower level control loops of the same magnitude as those in higher loops. In such a system there is little delegation of authority downward. Delegation of authority from higher to lower levels necessitates longer waits between commands from higher levels so that there is time for sequences of lower level commands to be executed and evaluated. In a simple linear control system with time constants of the same magnitude in inner and outer loops, the loops may be exchanged with no loss of meaning.

In human conditioning terms, a tight connection from teacher to learner is where the learner's muscle is continuously forced through a given response pattern by the teacher or teaching device. A somewhat looser connection is where the learner is set to track or continuously null the error between his own response and some ideal response. A looser connection still is where a series of discrete subgoals are set where the required response to get there is more or less obvious. An extreme of loose connectivity is where the goal is not at all specified in terms of required response, but where the human learner knows that a reward will eventually be forthcoming if after much free operant or trial and error behavior he happens to approximate the goal. Here the learner has authority to "try anything that works."

The future of computer-aided instruction will probably see increased attempts to apply formal theory, both deterministic and stochastic

or some combination. Any theory will require a better characterization of the set (or continuum) of mutually exclusive behavior categories--the "state space" of the learner-plus-controlled process. Deterministic theory demands a specification of the difference or differential equation according to which learner-plus-process transitions from one state to another--with or without instruction, reward or other input forcing. Stochastic theory demands a specification of the probabilities of transition as a function of starting state, new state, previous state(s), input, etc. In either case optimization of instructional technique demands a precise a priori statement of how the rewards of improved performance (learning) trade off against costs of training.

## BIBLIOGRAPHY


1. Coulson, J.E., Programmed Learning and Computer-Based Instruction, Wiley and Sons, 1962.

2. Suppes, P., Computer-Based Mathematics Instruction: the first year of the project. Institute for Mathematical Studies in the Social Sciences, Stnaford University, 1964.

3. Feurzeig, W. and D. Bobrow, Mentor: A Computer Language for Programmed Discourse, preliminary draft, mimeo, BBN, Cambridge, Massachusetts.

4. Bitzer, D.L., E.R. Lyman and J.A. Easley, Jr., The Uses of Plato: A Computer-Controlled Teaching System, Report R-268, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, Oct. 1965.

5. Maher, A. and Cook, L. Introduction to the Coursewriter Language, IBM 1410 Manual, 1964.

6. Smallwood, R., A Decision Structure for Teaching Machines, M.I.T. Press, Cambridge, Mass. 1962.

7. Logcher, R.D. et al, Structural Design Language, MAC Memorandum M-234, Sept. 15, 1965, Mass. Inst. of Technology., Department of Civil Engineering, Cambridge, Mass.

8. Rosenberg, R., Computer-Aided Teaching of Dynamic System Behavior, ESD-TR-66-260, USAF, Hanscom Field, Bedford, Mass.

9. Telcomp Manual, Bolt, Bernaek and Newman, Inc.- 50 Moulton Street, Cambridge, Mass.

10. Salkind, C.T., The Contest Problem Book, Random House, N.Y., 1961. [p.65, problem no. 29]

11. Bobrow, D.G., Natural Language Input for a Computer Problem Solving System, Ph.D. Thesis, M.I.T., Sept. 1964. (Available as Project MAC Report TR-1.

12. McCandlish, S.G., A Computer Simulation Experiment of Supervisory Control of Remote Manipulation, Engineering Projects Laboratory Report No. DSR 9960-2, M.I.T., Cambridge, Mass., June 1966.

UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R&D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

1. ORIGINATING ACTIVITY *(Corporate author)*
ENGINEERING PROJECTS LABORATORY
Department of Mechanical Engineering
Massachusetts Institute of Technology

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b GROUP
N/A

3. REPORT TITLE
Investigations in Computer-aided Instruction and Computer-aided Controls

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Report

5. AUTHOR(S) *(Last name, first name, initial)*
R. C. Roseberg T. B. Sheridan
S. G. McCandlish

6. REPORT DATE
April 1967

7a. TOTAL NO. OF PAGES
26

7b. NO. OF REFS
12

8a. CONTRACT OR GRANT NO.
AF 19(628)-3317

b. PROJECT NO.
7682

c.

d. Task 768204

9a. ORIGINATOR'S REPORT NUMBER(S)
ESD-TR-67-289

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
DSR 9960-4

10. AVAILABILITY/LIMITATION NOTICES
Distribution of the document is unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Decision Sciences Laboratory
Electronic Systems Division
L. G. Hansocom Field, Mass. 01730

13. ABSTRACT

This report is a summary of two research projects performed under USAF 19(628)-3317 in the area of computer-aided instruction and man-computer interaction. In 1965 Rosenberg completed a study entitled Computer Aided Teaching of Dynamic System Behavior published as ESD TR-66-260. This study demonstrated that within the delimited area of formal engineering theory a computer simulated laboratory could be built in which freshman students could pose problems, observe displays of machine responses, and thereby learn formal discipline with only minor interaction with a human teacher. An outline of Rosenberg's experiment and conclusions is given in Section I, along with some more general "Observations on the Use of Computers in Instruction," dealing with prospects for computerized tracking monitors. In June 1966 McCandlish completed a study entitled A Computer Simulation Experiment of Supervisory Control of Remote Manipulation, to be published concurrently with the present report as a separate ESD document. McCandlish summarizes how, for a formally well-defined task like grasping a block with a pair of jaws, removing the block from a hole and placing it in a second hole, the human can perform the task through the computer, but only with certain difficulties. He investigated such display rate, time delay between human responses and knowledge of results, and nature of command statements. An outline of McCandlish's experiment and conclusions is given in Section II, followed by "Observations on the Relation Between Computer-Aided-Instruction and Computer-Aided-Control," the latter being where the human serves as a supervisor and subgoal setter of a lower level semi-automatic system which interacts directly with the environment to accomplish a task.

DD FORM 1 JAN 64 1473

UNCLASSIFIED

Security Classification

Security Classification

| 14. KEY WORDS | LINK A ROLE | LINK A WT | LINK B ROLE | LINK B WT | LINK C ROLE | LINK C WT |
|---|---|---|---|---|---|---|
| | | | | | | |

## INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

GPO 886-551

Security Classification