

# PDP-8/LVE INTERACT system in an undergraduate environment

H. KEITH RODEWALD

Central Michigan University, Mt. Pleasant, Michigan 48858

This article outlines the advantages of and limitations on the use of a computer system in an undergraduate program in psychology. Illustrative material is drawn from the author's experience in a setting characterized by serious commitment to undergraduate education.

## SYSTEM CHARACTERISTICS

The system consists of three main pieces of equipment: a PDP-8/I 8K computer, an ASR-33 Teletype, and a Lehigh Valley Electronics INTERACT system. The 8K computer is sufficiently large to contain the ACT compiler (Millenson, 1971) and several complex programs. So far, computer storage space for programs is not a limitation on our use of the system. When the ACT compiler which is employed in on-line control of experiments is not required, Digital Equipment Corporation's FOCAL compiler, which is used for data analysis, is resident in the computer. Although the computer is about 90% committed to on-line control of experiments, its easy conversion to an 8K data-processing device employing a very simple language is an added benefit in our setting.

The ASR-33 is the main input/output device for the system. Programming in ACT and FOCAL is in conversational mode. One talks directly to the computer through the Teletype keyboard. The computer corrects most programming errors, as soon as they are committed, in either ACT or FOCAL languages by typing an error message when a faulty statement has been entered. The ASR-33 allows permanent storage of programs in both ACT and FOCAL languages. The device contains a punch and a reader for program tapes. Preparation of tapes is a simple one-stage process. The punch and read devices are quite slow. In settings which require frequent changing from ACT to FOCAL and feeding in of different programs, the slow tape reader is a definite limitation. We seldom experience much inconvenience resulting from input/output speed.

The INTERACT interface is a tremendous convenience in our setting. The principle application of the system is the control of experiments in a course in the stimulus

control of behavior. For this purpose, four two-key operant chambers with in-line projectors are in use. Since the interface contains four independent "stations," the four chambers can be run independently. A "station" consists of 11 stimulus output supplies, two 3-A and nine ¼-A sources. Eleven response input pens are available for recording -28-V pulses. The stations are independent in the sense that they can be programmed for different (in this case) schedules of reinforcement. For this kind of classroom use, the number of chambers could easily be doubled without adding hardware. To do so, one wires two operant chambers per station. This allows eight different schedules to be in effect simultaneously, but the chambers would be activated in pairs. There are only four stations, and, when one is activated, all schedules stored for that station are started. Pairing is a limitation only when students need to start an exercise at different times throughout the day. This can occur when some students are ahead of the normal progression through the set of exercises or when they desire additional computer time. Using one station per chamber is a matter of convenience in our application. Four stations running simultaneously on quite complex schedules have required only about two-thirds of the programming space in our 8K machine.

The data retrieval capability is sufficient for current purposes. It consists of counts and summations of responses and reinforcers occurring during specific elements of a schedule. Retrieval of time can also be obtained for purposes of latency and IRT measures. A limitation on data handling must be noted. Calculations based on incoming information cannot be performed. It is, therefore, not possible to alter a schedule by, say, evaluating a discrimination index and then making a decision to change a schedule parameter or stimulus configuration. This limitation can be circumvented in not too crude a fashion by retrieving data while the experiment is in progress and making an external evaluation. It is then possible to change parameters and stimulus configurations via the Teletype without actually disturbing

the schedule in effect. The lack of real-time data manipulation is the major limitation of the system. Of course, a cumulative recorder can be employed with considerable advantage in this kind of research setting, since its inspection can lead to a change in the schedule by flipping a switch, which causes the stored program to jump to another set of contingencies without disrupting the behavioral environment. The lack of what might be called total automation is not a major problem in our use. The power of the system as a control and recording device is so great that even this apparently major limitation is in practice not particularly important.

The data-manipulation problem is experienced only when the facility is on-line with a behavioral environment. At any other time, the DEC software can be used. Statistical computations of large variety have been performed on the system. The only problem in that application is the limited storage capacity of an 8K machine. This is circumvented by computing "on the fly." Essentially, this amounts to having the computer perform arithmetic on every piece of data at the time of input from the Teletype. Owing to the high speed of the computer, one can feed data into the machine about as fast as he can enter it in the keyboard. Using this approach to programming in the FOCAL language, several quite complex analyses of variance have been programmed. A student managed to generate a 10 x 10 correlation matrix from raw data with unlimited  $n$  with a few locations to spare.

## UNDERGRADUATE ENVIRONMENT

The system described above is devoted nearly 90% to use by undergraduates. The students range from second-semester freshmen to seniors. Their backgrounds in science and technology seem to bottom out at one course in high-school algebra. Most are interested in majoring in psychology and have a specific desire to concentrate somewhat in operant behavioral analysis. They have all taken an introductory course covering this approach to psychology. Given this set of constraints, the computer system was introduced into the undergraduate curriculum as a tool for the control of experiments.

There are several points of importance when considering equipment for undergraduate use. The interest in the computer is as a tool for further work in experimental psychology. This suggests that learning to use the tool should remain a part of the study of psychological principles and should not become, at this time, a

major study in itself. It seems that the smoothest way to get the student involved in computer control is via software which is directly related to aspects of their current competencies. Use of an Automatic Contingency Translator (ACT) represents a very small step from verbal descriptions of contingencies to on-line control. A rich scheme for the initial step is provided by the use of state diagrams (Snapper, Knapp, & Kushner, 1970). All the students are familiar with schedules of reinforcement, including multiple schedules. It is a simple matter to transfer this information to the drawing of state diagrams. Some students know the rudiments of Mechner notation (1959), and this makes the transfer even more probable.

The availability of an easy-to-use software package cannot be overemphasized in an undergraduate setting. What are the characteristics of "easy-to-use"? Going from idea to experiment is a rather complex chain of performances. The word "easy" indicates the need for an easily programmable approach to the development of this chain. The final element is already available: the students persist in talking about behavior with respect to schedules. One next requires them to write schedules in the form of state diagrams. This is accomplished in about 1 h; writing complex diagrams requires another hour.

Once the state diagram, which actually addresses behavior, is written, software is required. The ACT language allows the student to translate his state diagram into a common, indented outline of the kind familiar to him since elementary school. The outline is the format for typing the program into the computer. The ACT language has very few specific terms: IF when a transition is based on a response occurrence, AFTER for time, FOLLOWING for, say, a reinforcer, GOTO indicating the direction of transition. Quite complex diagrams and thus schedules can be stored in the computer by the average student after about 2 h of supervision.

Another aspect of "easy" is on-the-spot correction of errors. The ACT compiler recognizes virtually all errors a student is apt to commit in translating his diagrams and informs him immediately so that the error can be corrected. Since typing is, at best, a weakly developed skill for many students, much time is spent at the keyboard. If the program were completed before no-go information became available, the conditions for frustration for many students would soon be established. Neophyte FORTRAN programmers often

complain of long turn-around time required just to see if their program ran. Granted that FORTRAN is more complex than ACT, it is still reasonable to attribute some of the ease with which ACT is learned to a more desirable learning situation. FOCAL also has this immediate correction feature. What could be better for student and teacher than a tutor with infinite patience?

Given that 4-6 h of tuition, often done in laboratory sections of eight students, moves the student from slightly confused to completely debugged, some additional time must be spent on machine operation. This facet of training requires only a few minutes. It amounts to little more than turning on power switches. Wiring in the operant chambers is straightforward. One attaches leads from interface outputs to the in-line projector control panels and to the feeder control panel. Another lead from the response-key panel to an interface input completes the wiring. We have found that 6 h is sufficient to develop competency as a programmer-operator in our course. This includes taking the machine as a piece of junk and getting it up and running. Teaching assistants are recruited from students who have just completed the course.

In summary, the system described here is a powerful research instrument for graduate students and professionals. It is also an ideal device for moving undergraduate students to a point at which they can engage in nontrivial experimental techniques. This can be accomplished in the face of their concurrent activity in three or four other classes and other campus distractions. Having spent 2 weeks of the semester on technical details without the aversiveness of high-ratio requirements in unfamiliar territory, the student is ready for 13 weeks of laboratory work in the area of stimulus control. At this point in his education, the student has free access to the computer-controlled laboratory. By midsemester, the instructional staff's role as necessary agent in the daily running of the laboratory is successfully faded out. Since students can now use the laboratory without supervision, an acid test of our approach to preparing students can be made. Repairs resulting from student misuse over two semesters and about 50 students is in the neighborhood of \$50; this is competitive with faculty accidental misapplication of principles.

#### AN EXAMPLE OF APPLICATION

The instructor prepares program tapes for a series of exercises. Students do not actually work up tapes until very late in the semester. This

procedure results in uniformity of program requirements and, therefore, a more smoothly running laboratory during the performance of the required exercises. However, these instructor-prepared programs are given to the students and discussed in detail. Our experience has been that this procedure results in easy transition to completely independent programming work later in the semester and good performance on independent study projects in the next semester.

The prepared exercises are as follows: magazine training and keypeck shaping; adjusting and stabilizing an RI performance; establishing a multiple RI-RI using two stimuli consisting of red ground with three white dots in a vertical array and green ground with three horizontal dots; multiple RI-extinction with red-vertical as  $S^D$  and green-horizontal as  $S^\Delta$ ; test for attention to these stimulus components, as well as generalization on hue and array orientation, obtained by adding yellow and 45-deg array during the test for attention; simultaneous conjoint measurement of generalization on hue and orientation by presentation of the nine combinations of three background colors and three figures; development of a two-key performance; and development of a conditional discrimination using two keys and two colors of the houselight. The experimental animal which tolerates all of this in his first semester at college is a barnyard-reared chicken. Total costs for animals this semester is \$2.50. The cost is for food; farmers are willing to lend the birds at least until they reach marketing age.

The following state diagram and computer program is one of the required exercises. The exercise was used to replicate and extend the data obtained in Reynolds's (1961) study on attention. The wiring involved an in-line projector, feeder, key and E-initiated pulse. The outputs from the station were wired directly to the projector, using this scheme: S1 = red, S2 = yellow, S4 = green, S10 = vertical array of dots, S20 = 45-deg array, S40 = horizontal array, S1000 = feeder pulse. The inputs were R1 = key, R4 = E's switch supplied on the station itself. All other potential stimulus outputs, S2000, etc., were left blank. Above each output on the station is an indicator lamp which shows when that "stimulus" condition is in effect (See Fig. 1).

There are several specific points to note about this state diagram. A state or "stimulus" condition  $Sm.n$  means that output m is active; n is a counter of m states. State  $Sm.n$  has the stimulus characteristic of  $Sm.j$  but is functionally different from it.  $S0.m$

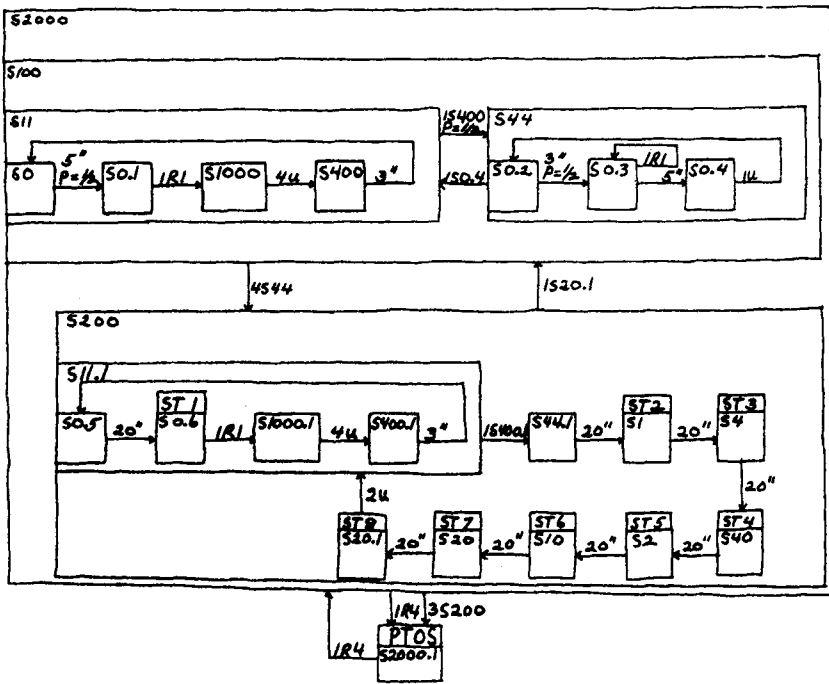


Fig. 1. State diagram employed in a study of attention (see text).

has no associated output; it is a null state used in programming. The condition  $x = y$  means that, upon completion of  $x$ , a transition will take place with probability  $y$ ; otherwise,  $x$  occurs again. The station output is a binary representation of an octal number,  $S_x$ . Therefore, S11 activates Outputs 10 and 1; S44 activates Outputs 40 and 4.  $u$  has the value of  $1/100$  sec. ST  $b$  means "store in Bin  $b$ " some value specified by the program. For instance, ST 2, which is initiated in S1, stores the number of responses made during the 20 sec of S44.1.

A functional analysis of the diagram is obtained by reading through it. The panel shows initial activity at Outputs 2000, 100, 10, and 1. After 5 sec, the probability of moving from S0 to S0.1 is  $1/2$ ; failure to move leaves S0 in effect for another 5 sec. When S0.1 comes into effect, a reinforcer is obtainable if a keypeck, R1, is emitted. S1000 output pulses the feeder and 4u later moves to S400. The indicator light above 400 on the station panel serves to inform an observer that grain is available to the animal. Three seconds later, the R1 is in effect again. S11 is red background with vertical dots superimposed, and this stimulus is  $S^D$ . When S44 is in effect, another random interval occurs, S0.2, during which R1 emissions are ignored. Once S0.3 is entered, R1 occurrences reset the timer in S0.3. Only after 5 sec of no responding can S0.4 be entered, and this state must occur before S11 is reinstated. This is a 5-sec protection contingency in  $S^A$ . Following completion of four  $S^A$

periods, the program moves from State S100 to State S200. In S200, data are collected; therefore, the light above the 200 output informs E that the data collection phase is in effect (see later). S200 samples  $S^D$  and  $S^A$  and their components, yellow ground and 45-deg array. Each store statement collects the R1 occurrences in the state which precedes it. Upon completion of the sample, indicated by the exit from S20.1, the multiple R1-extinction is once again in effect.

Data output is initiated in either of two ways. Following completion of three samplings or on an R4 pulse from E, the Teletype lists the stored information along with the station identification number. If the data come off following three samplings, S200s, they are listed as three blocks of eight values corresponding to the order in which they were collected. The serial position in a block identifies the values with the state which was in effect when they were collected. Here the first value corresponds to responding in  $S^D$ , the second to  $S^A$ , the third to red, etc. The importance of identifying sampling by use of a particular panel indicator light is now apparent: If the experiment must be interrupted by E, it should be done when sampling is not occurring; therefore, initiate output, R4, only when S200 is inactive.

In the present example, when data output is initiated, the organism is in time-out, since only Output 2000 is active and it does not supply the chamber. An R4 by E reinstates the schedule.

With this program, my students

```

S2000
S100
S11
S0
A5" GOP = 1/2S0.1
S0.1
I1R1GOS1000
S1000
A4UGOS400
S400
A3" GOS0
F1S400 GOP = 1/2S44
S44
S0.2
A3" GOP = 1/2S0.3
S0.3
I1R1GOS0.3
A5" GOS0.4
S0.4
A1UGOS0.2
F1S0.4GOS11
F4S44GOS200
S200
S11.1
S0.5
A20" GOS0.6
S0.6 [STORE 1]
I1R1GOS1000.1
S1000.1
A4UGOS400.1
S400.1
A3" GOS0.5
F1S400.1GOS44.1
S44.1
A20" GOS1
S1 [STORE 2]
A20" GOS4
S4 [STORE 3]
A20" GOS40
S40 [STORE 4]
A20" GOS2
S2 [STORE 5]
A20" GOS10
S10 [STORE 6]
A20" GOS20
S20 [STORE 7]
A20" GOS20.1
S20.1 [STORE 8]
A2UGOS11.1
F1S20.1GOS100
F3S200GOS2000.1
I1R4GOS2000.1
S2000.1 [PRINT OS]
I1R4GOS2000
$$
-----
IN S0.5:R1 COUNT
IN S44.1:R1 COUNT
IN S1:R1 COUNT
IN S4:R1 COUNT
IN S40:R1 COUNT
IN S2:R1 COUNT
IN S10:R1 COUNT
IN S20:R1 COUNT
$

```

Fig. 2. Computer program corresponding to the state diagram in Fig. 1 (see text).

have replicated Reynolds's findings. The birds were, to the lady, color responders. Further, stimulus-generalization gradients of nonzero slope were obtained only on the attended-to dimension. The next exercise in the series measured responding to the nine combinations of the three orientations and the three

colors of background. The program required only the addition of another state in S200. The two studies led to a discussion of data analysis and the problems of finding convenient metrics for describing the birds' ordering of stimulus properties. In passing, it should be noted that while the INTERACT system is ideal for operant research, retrieval of data in forms leading to more theoretical treatments is easily achieved. Here we deal lightly with conjoint measurement (Krantz, Luce, Suppes, & Tversky, 1971).

The direct translation of state diagram to computer program is seen in the program below. The level of indentation corresponds to the level of nesting of the states in the diagram. This program requires about 16% of our programming capacity.

#### SHARING

The time required for class use is actually quite short. It is scheduled routinely and occurs only during the normal school day. This suggests the possibility of sharing the facility with other users. While that could be done, it involves several special considerations.

One of the characteristics of our use is the allowance for free time. Students often run experiments during unscheduled use of the machine. Sharing would require some curtailment of this, from an educational point of view, highly desirable student option. Scheduled free periods can be employed as a

remedy for this problem, but one result is empty scheduled time. Effective free-use schedules are difficult to engineer when student class schedules are complex. A second aspect of scheduled free use is seen in regard to the low-speed input/output device. Switching from ACT to FOCAL may require as much as 30 min. Several changes during the day guarantees 2 or more hours of down time. Our solution has been to load ACT early in the day and to load FOCAL for use in the evening. This limits both ACT and FOCAL users and sometimes works to the disadvantage of some busy, talented students. The best solution in the context is to add a high-speed reader to the system, which would reduce loading times to a few seconds. Under no conditions could the system be used for statistical manipulation and on-line control at the same time.

Use of our system by several workers is permitted in one-line application with ACT. In fact, we try to keep one station unscheduled so that students taking independent study or directed research can get on-line any time ACT is running. One, so far unimportant, issue can arise in this situation. We have only one input/output device. The input function has priority; therefore, output is inhibited any time programming of a station is occurring. With a low-speed reader, the time required for this function is not trivial during high-density usage. The ideal solution is to add a Teletype which is

devoted to data printout. Two options are available: add an output Teletype or add another input/output Teletype. The latter is preferable. While ACT is running, the second Teletype could serve only as an output device. When FOCAL is running, the second device could be used as both input and output in a DEC time-sharing scheme.

The addition of these two pieces of hardware would enable more efficient usage of our basic configuration. The gain in efficiency would be felt within the department. It is doubtful that our application would tolerate interdepartmental sharing. Of course, the late evening hours could be set aside for others to use the facility, but this is not equivalent to the multiprogramming time-sharing capability that is ideal for a shared facility.

#### REFERENCES

- KRANTZ, D. H., LUCE, R. D., SUPPES, P., & TVERSKY, A. *Foundations of measurement*. Vol. 1. *Additive and polynomial representations*. New York: Academic Press, 1971. Pp. 245-368.
- MILLENSON, J. R. A programming language for on-line control of psychological experiments. *Behavioral Science*, 1971, 16, 248-256.
- REYNOLDS, G. S. Attention in the pigeon. *Journal of the Experimental Analysis of Behavior*, 1961, 4, 203-208.
- SNAPPER, A. G., KNAPP, J. Z., & KUSHNER, H. K. Mathematical description of schedules of reinforcement. In W. N. Schoenfeld and J. Farmer (Eds.), *Theories of reinforcement schedules*. New York: Appleton-Century-Crofts, 1970. Pp. 247-275.