Rapid response and flexible experimental control with a small on-line computer: PSYCLE*

D. DOUGLAS CREELMAN
University of Toronto†, Toronto, Ontario, Canada

This is an outline of some solutions to problems in using a relatively slow computer, without expensive peripheral apparatus, for control of psychoacoustic experiments. Major concerns were accurate timing and the ability to change from one experiment to another without extended reading of programs from paper tape.

The PSYCLE system, consisting of separate compiler and operating systems, is currently running on a Digital Equipment Company PDP-8/S computer in our Auditory Psychophysics Laboratory. The computer has 4K of 12-bit memory, with input-output through a slow-speed (10 cps) Teletype and paper-tape reader/punch. Figure 1 is a diagram of the configuration. The objective in developing PSYCLE was to make this minimal configuration capable of running psychoacoustic experiments in which timing of auditory signal presentations was critical. A further objective was to make the system flexible, allowing rapid setup of new experiments without lengthy reading of programs from paper tape into core. Since a number of Es use the laboratory each day, each with a different set of experimental conditions, such flexibility was imperative. This paper describes our solutions to these two problems and outlines some general principles which we have come to feel are important for such systems.

RAPID RESPONSE AND ACCURATE TIMING

In comparison with currently available machines, the PDP-8/S operates at a snail's pace. To control events to within fractions of a millisecond in an environment where single machine instructions can take that long is a considerable challenge. In newer machines, the problem is not quite so critical, but it still exists.

*Supported by the Canadian National Research Council under Operating Grant NRC-APB-111 and Major Equipment Grants NRC-APA-1 and -APA-7. The special help of J. L. Shifren, Z. Jordanoff, C. L. Laker, and N. C. Copperman is gratefully acknowledged.

†An extension of the PSYCLE concept is currently under development for the PDP-9 at the Defense Research Establishment Toronto by V. K. Taylor and C. Mains, with some help from the author. The development of the programs described here benefited from discussions with Dr. M. M. Taylor at DRET.

Where timing of stimulus events is important, the events must be precisely predictable. The PSYCLE operating system allows this control.

Our experiments, in common with many in psychophysics, involve repeated sequences of trials, each broken down into a series of events involving information lights for the Os, presentation of signals according to some random schedule, recording of responses (of the sort which can be signified with a pushbutton), and feedback to the Os regarding the outcome of each trial. Events within each trial, and the succession of trials, must be precisely timed.

Central to the system is a down-counting clock which is part of the interface. The clock can be loaded and started under program control and thereafter will continue to run on its own. The output from the clock is connected to the computer's single interrupt and, when experiments are

running, no other devices are able to cause an interrupt. Markowitz and Nickerson (1968) have discussed alternative approaches to clocks; in this context the external clock is the only possible solution. Figure 2 is a simplified flow diagram of the events which occur immediately upon an interrupt from the clock.

When a time interval is completed, a preset sequence of instructions is executed. First, the clock is reloaded with a previously selected time, corresponding to the next event. Immediately after this, a sequence of instructions, mostly input-output commands, are executed. These instructions cause the interface to present stimuli, light lights in the Os' booths, or read responses into buffers from the Os' response buttons. Then, still as part of the instructions specific to this particular time period, any further operations are done, such as making random selections necessary for the next event time or setting the apparatus to new values of stimulus parameters. This set of event-specific machine instructions terminates with a transfer of control to the operating system (OS), which does the executive work necessary to prepare for the impending new clock interrupt.

The OS has no direct control of experimental apparatus or of the clock. Its function is only to prepare the interrupt-handling section. The OS determines the next clock time after the currently running one and installs that value in the appropriate place; it decides which set of interval-specific instructions is to be executed next and organizes them for immediate

Fig. 1. Simplified block diagram of computer-controlled Auditory Psychophysics Laboratory. To the right are sound-attenuated experimental rooms for human Os. The interface and audio apparatus in the center occupies three 19-in. racks, while the computer is a table-top model.

Fig. 2. Sequence of events at the beginning of each time interval within an experimental trial, under control of the PSYCLE operating system. Each experiment has its own set of instructions and timing, generated by the PSYCLE compiler and saved for later use when experiments are run.

execution when the time comes. Finally, the OS sensitizes the interrupt which had been automatically disengaged so that none of this activity could be interrupted by the clock before everything was ready. There is, of course, a check to make sure the clock was not already calling for attention before the OS was ready for it, and if so a trap is executed.

By enabling the clock interrupt only after the OS is ready, we avoid the necessity of taking time after each interrupt to save currently active registers or checking flags to identify the source of the interrupt. This is at variance with normal procedure. It is necessary, because such saving and checking takes precious time before output can begin and introduces the probability of different numbers of instructions being executed on the occurrence of an interrupt from the clock. We then would have an unknown time in the sequence of an experiment trial.

The actual mechanics of the preparation for succeeding sets of instructions by the OS has taken different forms in successive versions of PSYCLE. In an earlier version, a block of instructions was actually

physically copied from one place in memory, where it resided permanently, into a set of contiguous locations immediately below the core location to which control is transferred by an interrupt. In our machine this transfer on occasion took too long, and so we resorted simply to letting the OS organize a pointer to the appropriate location in the permanent list. This was done at the cost of an added indirect transfer instruction which is executed after each interrupt before the actual output can begin. Each such instruction adds a constant to every time interval which an E thinks he has specified, and must be taken into account.

FLEXIBILITY AND RAPID CHANGEOVER

One of the important advantages of controlling experiments with an on-line computer is generality—but with generality comes the problem of reprogramming for each new experiment. Such changes can be done easily if one is rich in peripheral equipment and computer memory. Haber (1968), for instance, apparently uses magnetic DECTape to store new sets of conditions for his experiments which are controlled by a PDP-8. The system programs to retrieve data from the tape take a goodly amount of core storage, and the initial cost of a DECTape unit along with the necessary hardware controller is more than a small-scale user might be expected to put forth. One important objective of the PSYCLE system was to make reprogramming feasible from a slow-speed paper-tape reader.

The above outline of the OS points toward a way of doing this. Experiments differ in the particular events which happen within each trial (mostly input-output instructions), and in the timing of these events. The numbers of trials to be run and alternatives for random selection within each trial also change from one experiment to another. The PSYCLE compiler was designed, then, to take a description of an experiment, decode the source language, and produce a short paper tape. The tape contains information which is specific to the experiment. Programs can be compiled well before the experiments are actually run. At running time, the compiler output is read into the computer, and the logical setup for the experiment is ready to go. When the E has plugged together his own configuration of the audio apparatus, the day's run can start. To change experimental conditions, he loads a different previously prepared paper tape which installs a new set of instructions for the apparatus in place

of the set currently operating.

The larger system needs only to be loaded into the computer memory once, at the beginning of the day. Like most large system programs, this takes more than 20 min, while changes from one experiment to another, once the larger system is present, take only 15-20 sec. In addition to this convenience, the compiler and the OS programs both can be in memory simultaneously. This makes it possible for an E to write new experimental programs, changing timing or introducing new conditions, on the spot if he wishes to.

SOME CHARACTERISTICS OF THE PSYCLE LANGUAGE

The basic unit of the language is a line of text. A line starts with a time (in milliseconds), followed by a list of the events which are to start at the beginning of that interval and continue for its duration. In our lab, the available events include switching a function generator on or off, setting any of three double-throw electronic audio switches, stepping any one of three attenuators up or down, and controlling any of five lights in each of three listening booths.

A special category of event is random selections made among sets of alternative possible events. Up to four independent random selections can be made during one time period, and the results (or their inverses) used later during the same trial as often as required. Data can be recorded from each of three Os separately or from the three simultaneously. The data are saved sequentially in memory and are punched onto paper tape at the end of each run. The length of the run, the number of repeated trials, is specified in the program. The E can call for special events to happen only before the run begins or only after it has ended.

A much-used capability is an instruction which transfers control to a special subroutine which the E has written to do a job not available within the OS. There are sufficient spare memory areas for setting up even rather extensive programs. As an example, at the moment the programs to run the adaptive PEST psychophysical procedure (Taylor & Creelman, 1967) are invoked in this way. This procedure considers data from each O on a trial-by-trial basis, and sets stimulus levels for succeeding trials in an attempt to find that stimulus level for each which yields a selected level of performance.

SOME CONCLUSIONS

Small computers can be used in the laboratory for experimental control without investment in expensive

high-speed input-output or bulk-storage peripheral apparatus. Most behavioral applications require precise timing of events, but the times involved are relatively long (milliseconds, at least) compared to the computing speed of even relatively slow machines. Our solution has been to use an external clock to keep track

of time while the computer performs executive functions and to develop an off-line compiler to organize the different experiments into a shortened form for use in the actual running.

REFERENCES

HABER, R. N. An on-line computer in a

visual perception laboratory. *Behavior Research Methods & Instrumentation*, 1968, 1, 86-93.

MARKOWITZ, J., & NICKERSON, R. S. On timing events in computer-controlled experiments. *Behavior Research Methods & Instrumentation*, 1968, 1, 82-86.

TAYLOR, M. M., & CREELMAN, C. D. PEST: Efficient estimates on probability functions. *Journal of the Acoustical Society of America*, 1967, 41, 782-787.