H. A. Reich

An Experimental System for Time-shared, On-line Data Acquisition

Abstract: With the increasing availability of terminal-oriented, time-shared computers, it now becomes feasible to extend the method of use to include real-time, on-line data acquisition and data reduction. Here described is a particular embodiment of such a system, using an IBM 1050 terminal and the IBM Research M44/44X and the APL/360 Model 50 computers. Reasons leading to the choice of equipment, special devices, programming considerations, data rates and some economic factors are considered.

Introduction

Large general-purpose, time-shared computers with a wide variety of languages are becoming commercially available at the present time in many cities.[1] Private time-shared systems exist also at a number of universities, the MAC system at the Massachusetts Institute of Technology being one of the first and most notable. Private time-shared computer systems are also in use in some large industries on a routine basis. At the same time, automatic data acquisition and on-line data reduction have been widely discussed,[2,3] although not in the context of a terminal-oriented system. Recently there has been an announcement of commercial services which encompass a variety of data acquisition methods, together with proprietary data reduction programs. It therefore seems to be of interest to see whether a large general-purpose, time-shared system is adaptable to the methods of automatic data acquisition and data reduction.

We describe here our experience in using two widely different computers, the M44/44X using FORTRAN[4] and an IBM System/360 Model 50[5] employing APL/360, both located some 35 miles from the laboratory. The computers were accessed over the standard voice-grade telephone network. For the present purposes we regard the two computers as equivalent and interchangeable.

Since a principal aim is to achieve low cost, and simplify the writing of data reduction programs, we exclude from consideration small and medium sized dedicated computers. These machines serve a different purpose, generally require much more programming effort, and have the ability to accept high data rates and also control the experiment.

The author is at IBM Watson Laboratory, Columbia University, New York, New York.

Application

We discuss here those aspects of the experimental design which are influenced by the slow data rate of the terminal. This is typically 15 characters/sec, although some terminals are slower. This limitation arises from the mechanical operations of the printer, and furthermore is enforced by the design of the general-purpose, time-shared system. Thus if faster terminals become available, and faster data rates are accepted by the computer, larger or faster experiments may be undertaken. The experimental apparatus was a pulsed nuclear magnetic resonance spectrometer, used to investigate various relaxation times occuring in a sample of solid helium-3. Since the experiment involves cryogenic temperatures, which can be maintained only for limited times, there is a premium on obtaining the maximum possible amount of information in one run, usually of six hours duration.

One datum consists of one measurement of the time interval between two pulses generated in a pulse transmitter, and two measurements of pulse heights, characteristic of the magnetism of the sample. The variation of magnetism with time generally follows a functional form consisting of a simple exponential, with sometimes the sum of two appearing. A set of data would then consist of say, 20 points along the relaxation curve, where the information of interest is the characteristic relaxation time, and several incidental instrumental parameters determined from the same curve. Each of the measurements is performed by a digital counter which measures, in turn, time and the dc voltage developed in an analogue gated integrator by the magnetism of the sample. The time is selected by setting switches by hand, and the entire sequence is initiated by manually pressing a start button. Since the characteristic relaxation times vary from 10^{-3} to 10^{+3} or more seconds,

depending on the sample temperature, density, frequency, and impurity content, we may wish to operate in such a manner that calculation, by nonlinear least squares, may be done after acquiring any number of data points. In this way, the experimenter is guided in the selection of relevant time intervals in order to make a 1% determination of relaxation time. This portion of the apparatus has been previously described.[6]

To generalize, then, the experiment is one in which

a) the experimenter can record in his notebook all the parameters, instrument readings, and results of his experiment, or

b) the output of the experiment is from a rather slow strip chart recorder attached to the spectrometer, or

c) the output of more sophisticated instruments, such as counters, digital voltmeters, or even multichannel pulse height analyser might be recorded.

In order to justify the development and operational expense of an automated system, several criteria have appeared to be relevant: An amount of calculation might have to be done sufficient to keep a person busy full time. Or keypunching might have to be done for later entry to a batch processing computer. In addition, the experiment has to be highly repetitive, so that the expense of writing data reduction programs can be recovered.

If the above conditions are fulfilled, it had been customary in the past to set up an automatic recording system, built around either card or paper tape punches. Card punches seem to have been preferred, in spite of the increased cost, because of the ease of reading and editing the information. Cards were also the medium on which programs were filed, for the same reasons.

An interface adapter is usually required to couple an experiment to a card punch. Several functions are required; BCD output of counters has to be translated to Hollerith code and scanned column by column into the card; contact closures may have to be sensed; apparatus parameters may have to be entered; all under the control of the interface adapter. Since many of the card punches typically used for these applications operate at 15–20 char/sec, most such applications may be coupled directly to a terminal with little loss in speed.

A terminal oriented system as described here completely does away with the intermediate storage medium. Programs are written, stored, debugged, and executed directly from the terminal. Data are entered directly and automatically from the terminal into computer storage, where they may be stored indefinitely, or processed and analyzed as they are received. The terminal printer thus serves as the laboratory notebook, having a record of all pertinent conditions, data and results computed.

The advantages of on-line operation are greater than might appear just from the decrease of turn-around time. The possibility of errors in transcription is eliminated. Errors in procedure become immediately apparent, so that they may be corrected during a run, before large amounts of useless data accumulate. Finally, the ability to relate data to physical concepts, such as the expected exponential behavior of some quantity with the reciprocal temperature, leads to immediate checks on theoretical understanding of phenomena. Novel or unexpected behavior may thus be immediately recognized and pursued.

The fact that data may be stored independently of the data reduction program gives rise to a further flexibility, in that if an unexpected functional relation is suspected, new analysis programs may be written, and data may be analyzed by someone else at another terminal simultaneously, and even remotely.

Choice of terminal

At least three terminals are widely available, and new ones appear continually. Therefore the remarks here are meant only to illustrate considerations leading to a choice. One might choose from:

a) *Teletype.*[7] Several models are available. Printer output, keyboard, and punched paper tape models exist. Data rates are of the order of 10 char/sec. Commercial interface couplers exist.

b) *IBM 2741.*[8] This is the smallest IBM terminal. Keyboard and printer are one logical unit. Speed is 14.8 char/sec. Auxiliary input or output equipment is not readily available, but may be attached by simple modification of the terminal. No commercial interface couplers exist but, again, adaptations are easily accomplished. There are two standard 7-bit codes.

c) *IBM 1050.* This is a somewhat more elaborate terminal, designed for multiple modular input and output units, and capable of having several terminals on one telephone line. The input and output devices may be selected by appropriate switches on a panel, or may be controlled by the remote computer.

For several reasons a 1050 terminal was chosen for the first implementation: 1) Modular design makes easy attachment of experimental input and output devices. 2) SMS technology (Standard Modular System, a system of discrete current elements mounted on cards) is easy to modify, because of discrete components and low packing densities. 3) Availability of card punches and readers gives the hesitant experimenter fall-back capability in case of computer unavailability due to malfunction, schedule, or saturation. Experience with the M44/44X system, which was available only four hours daily, justified this choice. The most important factor in time-sharing systems is the number of continuously available hours. With one programming system, 18 hours daily are common; no data have ever been lost in a period of $1\frac{1}{2}$ years; machine mal-

functions were sufficiently rare so that the 1050 terminal has been replaced by a 2741.

Figure 1 shows the arrangement of the terminal to form part of a data acquisition system.

Special devices

An experimental Data Interface Adapter (DIA) was designed and constructed to couple the output of the experiment, read in this case by a Beckman 6147 counter,[9] to the 1050 terminal. (Minor modifications enable a 2741 terminal to be used in an identical way.) The counter has an 8-digit BCD coded output and can be externally programmed by means of relays which select appropriate functions. A voltage-frequency converter is part of the counter, so that dc voltages, as well as duration or frequency may be recorded. The counter serves as a form of buffer store, since it keeps its count until read out and reset.

The DIA consists basically of two shift registers, and appropriate gating and clocking circuits. Register A provides 16 program steps, which control counter function and display format. Register B controls the scanning of the digits from the counter. It may be initiated at an arbitrary step, so that successive fields may have various lengths. Only one translating network is needed, which is easily changed to accommodate different voltage levels or codes. See Fig. 2.

Under control of register A, then, the DIA scans the BCD output of the counter, starting from a particular digit. Translation is first to one-out-of-ten, so that decimal switches or other contact closures may be sensed, and then to 7-bit EBCDIC code. Insertion of space and control characters required by the time-sharing system is accomplished by pulsing appropriate emitters under control of the A register. Selection of counter function, output format, and nonprinting control characters is controlled by means of a plugboard. This makes for great flexibility, as the plugboards may be prewired for different experiments.

The first DIA was constructed from components and cards found in the 1050 terminal. Here SMS technology was used, since the desired configuration was not apparent at the beginning of the project. Further replications would be better made from integrated circuits, since many of the logic functions, such as shift registers, can be obtained on one chip. Approximately 80 SMS cards were used in the first model.

The DIA is controlled from and interlocked with both the experiment and the remote computer. It waits for a signal from the counter which tells it that the counter is ready to be read, and upon receipt of a similar signal from the computer that it is ready to accept information, scanning starts. There is thus a variable time delay imposed upon the experimenter, due to the random response time of time-shared systems. Buffers are hence required to store information from the experiment until it can be scanned.

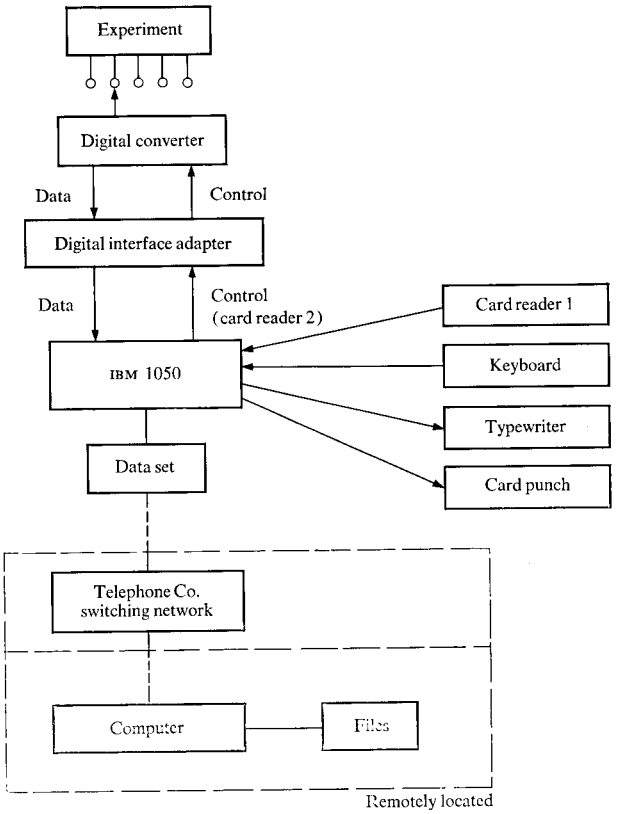

Figure 1 Diagram showing how terminal is arranged to form part of data acquisition system.

In our case the counter performs part of this function. Another buffer is in the form of a gated dc integrator which samples the signal and holds a dc voltage until reset some seconds later after read out. If real-time accumulation is required, the experiment must be interlocked until scanning can begin. At that time, successive measurements can be made, digitized and transmitted, up to the maximum data rate of the system, and to an extent limited only by the buffer size of the computer. In the APL system, for example, this size is 600 characters.

The design of the DIA is also influenced by the programming language used by the main computer. For example, in FORTRAN, successive numbers can be run together without intervening blanks. The format statement untangles the digits, and blanks are useful only for legibility. If the APL language is used, however, blanks or commas between successive numbers are almost mandatory, as then the numbers are interpreted as components of a vector. If no provision is made to insert punctuation, it is necessary for the program to unscramble the data. The DIA must also emit the control characters required by the computer to signify the end-of-message sequence. Since this varies from one computer to the next, and from one terminal to

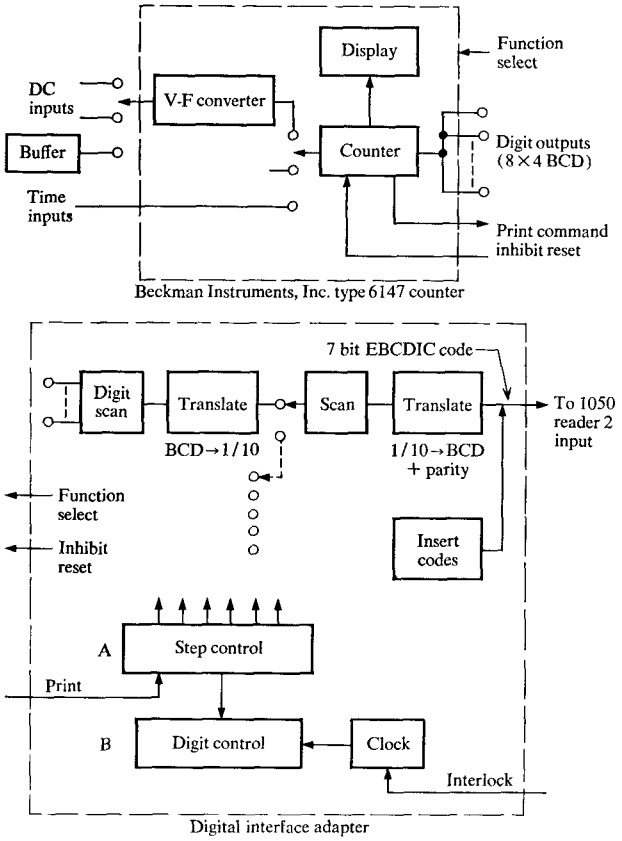

Figure 2 Digital Interface Adaptor.

the next, provision is made to emit *any* arbitrary bit pattern. These characters are controlled by plugboard wiring, so that speaking to a different time-shared computer involves at most the change of the board.

Programming considerations

In the past, with batch processing of numerical data, it was of minor consequence if a blunder appeared in the input. The job would result in an arithmetic or logical error, and execution would be terminated with an appropriate error message. The error could then be corrected and the job resubmitted. Only the turn-around time, several hours, would be wasted.

In the on-line situation, however, an unforeseen blunder can produce program failure. Since there no longer is a real record (i.e., cards) of the course of an experiment, an error which results in suspension of execution of a program could result in the loss of all data gathered to that point. It is therefore of some importance to construct the program so as to make this error unlikely. Another difficulty, not present with batch processing, would be computer down time. Computer operators have instructions in some cases to clear the machine memory when restarting. This also can result in the loss of data and has to be guarded against in a way unique to each computer.

In our particular experimental situation, collection of a set of data points takes perhaps 10 minutes, after which computation lasting a few seconds is performed. An independent measurement can then be started. If the computer fails, 10 minutes of data at most are lost. In the APL system, telephone communications can be interrupted, or less frequently, can transmit incorrectly, but loss of the line results in automatic saving of programs and results to that point. Ten minutes of experimental time are easily repeated in our case, and thus no special precautions, such as writing on disks or tapes, are employed.

If the computer fails, data taking could be continued on cards, with subsequent reduction. In practice, the advantages of immediate data reduction are so great compared to off-line reduction that no experimentation is attempted in the absence of the computer.

The data reduction program consists of several sections, each selected by a control section. The control section reads in data and control words, and analyzes what it receives to distinguish between control information and data. Control words (generally one digit), inform the data reduction program what type of experiment is being performed, what functional relationship to use in analysis, whether to plot the output on the terminal typewriter, or whether to use previously stored data or accumulate fresh data. This section is easily expanded to accommodate new functions.

Data words, upon being entered, are accumulated in a matrix. Upon receiving a predetermined number of points, or upon command, the matrix is stored externally to the program, and computation is performed. The first step in computation is examination of the data for out-of-range numbers, missing numbers, inconsistent data, exclusion of points that do not lie on an approximately smooth curve, etc. Use is made of all *a priori* knowledge about the data.

Least-square curve fitting is then done, using nonlinear iterative methods. After the "best" values of the parameters are determined, the program goes back and compares each residual with the average expected residual. If a point lies more than three RMS deviations away from the calculated curve, it is considered to be a mistake and is eliminated. The least-squares reduction is then repeated with the remaining points. After no more points are thrown out, a single line of output results. This output contains the values of the parameters, their standard error, the sum of the squares of the residuals, and a list of the points thrown out, together with an indication of why. The entire computation takes about 2 seconds on the M44/44X, which is less than the time for the typewriter carriage to return the width of a page. The M44/44X computer employed a compiled FORTRAN program.

APL data reduction proceeds similarly, but is slower, taking sometimes as long as 60 seconds. This interval is required because it operates as an interpreter and because the computer is shared by as many as 60 users at times. The APL system makes a separate charge for connection time and CPU time. This results in arranging the data reduction program so that data accumulation is in a very small loop, and computation is done only when necessary. This procedure would of course vary, depending on the billing method of the basic computer, in order to minimize the total cost. For example, a FORTRAN using computer that charges a fixed amount per connection hour could be programmed to carry out the least-squares reduction after each data point is added, thus enabling the experimenter to stop when he is satisfied with the precision of the result. APL appears to be as useful as FORTRAN.

Summary

The results indicate that, with reasonable economy, one can acquire data semiautomatically and can transmit this easily to a time-shared computer and obtain real-time data reduction. Writing of system and control programs is completely avoided. This system is not the same as a process control computer, because no provision is made for feedback. All the system output, be it results of computation, instructions to the operator, error messages, input data, etc., appears on the terminal typewriter.

Acknowledgment

I would like to thank L. Kreighbaum of the Thomas J. Watson Research Center for the design and construction of the data interface adapter. I would also like to thank the staff of the M44/44X computer for introducing me to time sharing and for answering innumerable questions.

References

1. New computer utilities appear periodically. See for example, *Datamation,* **13,** 103 (June 1967).
2. *Scientific American,* **215,** 128 (Sept. 1966).
3. S. J. Lindenbaum, in *Ann. Rev. Nuclear Science* **16,** 619 (1966). H. M. Gladney, in *Journal of Computational Physics* **2,** 255 (1968).
4. R. W. O'Neil in *Proc. Spring Joint Computer Conf.,* 1967.
5. K. E. Iverson, *Elementary Functions: An Algorithmic Treatment,* Science Research Associates, Chicago, Illinois, 1966.
6. H. A. Reich, *Physical Review* **129,** 630 (1963).
7. Teletype Corp., Skokie, Illinois.
8. International Business Machines Corp., Armonk, N. Y.
9. Beckman Instruments, Richmond, California.

Received August 8, 1968