

# A Time-Shared Computer System for Diffractometer Control\*

H. A. Alperin

Naval Ordnance Laboratory, White Oak, Maryland 20910

and

E. Prince

Center for Radiation Research, National Bureau of Standards, Washington, D.C. 20234

(July 27, 1970)

A system is described for controlling data acquisition, and for online data reduction, on up to eight neutron and x-ray diffractometers. The system uses a medium-sized computer, with the individual instruments sharing time. Storage of programs, data, and the intermediate results of computations on a rapid-access disk makes roughly 12K of the 16K core memory available to each user in turn for computations. All users' programs are written in FORTRAN. Each user has independent access to the computer, through his own separate typewriter, for input of control parameters and output of sample results. Final output data may be recorded on magnetic tape for permanent filing or for processing offline by a large computer.

Key words: Automatic control; computer control; diffractometer; FORTRAN; neutron diffractometer; real-time data processing; time sharing; x-ray diffractometer.

## 1. Introduction

During the past 10 years, a number of instruments have been built for the automatic and semiautomatic collection of single-crystal diffraction data, involving the diffraction of both x rays and neutrons. The earlier instruments [Prince and Abrahams, 1959; Abrahams, 1962] were mostly semiautomatic, with input parameters computed on an offline digital computer and transferred to a punched paper tape, which served as a program for the collection of data. The output data would also be recorded on punched tape, to be reduced to integrated intensities and structure amplitudes by the computer. More recently a number of instruments have been built [Busing, Ellison, Levy, King, and Roseberry, 1968; Beaucage, Kelly, Ophir, Rankowitz, Spinrad, and Van Norton, 1966; Fitzwater, 1965; Kruger and Dimmler, 1964] in which a computer was attached directly to a diffractometer, and served to compute settings, acquire the intensity data, and reduce the data to a greater or lesser extent online and in real time.

In controlling diffractometers directly by computers, there have been two distinct approaches. The first, and so far the more common, approach is to use a small

computer for each diffractometer instrument. A typical example of this type of system is the one described by Busing et al. [1968]. The other system, which is economical only if there are several diffractometers located physically close together, as is frequently the case with neutron diffractometers at a reactor, is to share time on a large computer. Previous examples of this approach are the system constructed at Brookhaven National Laboratory [Beaucage et al., 1966], at Ames Laboratory [Fitzwater, 1965], and at the Kernforschungszentrum in Karlsruhe [Kruger and Dimmler, 1964]. Each approach has its advantages and disadvantages. In the case of the small computer for each diffractometer, there is the advantage of complete freedom from interactions with users of other instruments in the same installation, but the computing capacity of the small computer is limited. In practice the real-time capability is restricted to control and basic reduction of data, and the data handling itself is limited. The time-sharing approach gives much greater computing capability to each user, the advantage of sharing expensive peripheral equipment, and vastly greater data handling ability, but requires an extensive system program to control the time sharing. However, it must be recognized that even these latter advantages may be effectively lost if the writing, debugging, and modification of the programs that are to be written by the practicing solid-state physicist or structural

\*A 10-minute 16-mm sound movie illustrating some aspects of this system may be obtained on free loan by addressing a request to Office of Technical Information and Publications, National Bureau of Standards, Washington, D.C. 20234.

chemist involve special computer knowledge or are inconvenient in any way.

At a research installation such as a reactor, there is usually a wide variety of different types of experiment in progress at any one time. These may include single-crystal structure work, powder diffraction studies, diffraction studies of liquids or amorphous solids, neutron inelastic scattering studies of phonon and magnon dispersion curves, or polarized neutron studies of magnetic materials. Although the apparatus for all of these experiments is similar, the programs for controlling the experiments and analyzing the data may be very different, and the experimenter's ideas of the proper procedure may change as experience accumulates. In order to achieve the necessary versatility and flexibility in a system for controlling diffractometers at the National Bureau of Standards Reactor, it was decided to use the time-sharing approach, with two particular design objectives. First, the individual instruments must have, as much as practicable, truly independent access to the computer; and second, it must be possible for virtually all user programs to be written in FORTRAN compiler language. In order to make the most effective use of available funds, it was further decided to design the system so as to hold capital cost to a minimum, at the expense of some increase in the complexity of the operating system program; and operating costs were to be minimized by designing a system which required no operator to run it.

This paper describes the system which has been developed to meet these objectives. The emphasis is placed on the general ideas behind the system design and, equally important, on the response of the users in terms of some of the features that have been incorporated into the programs written so far.

## 2. Computer Hardware Configuration

The NBS diffractometer control system uses, for its central processing unit, an XDS 925 computer,<sup>1</sup> with a core memory capacity of 16,384 24-bit words, and a cycle time of 1.75  $\mu$ s. The core memory is supplemented by a rapid-access disk memory with a capacity of 262,144 words and an average access time of 17 ms. Other peripheral devices include a console typewriter, a paper tape reader and punch, a card reader, and a magnetic tape transport. Figure 1 shows a block diagram of the computer configuration.

The computer communicates with the diffractometers and the experimenters by means of a system interface designed and built to NBS specifications by Xerox Data Systems, Inc. The system interface includes a teletypewriter interface, a master controller, and a station controller for each diffractometer. The teletypewriter interface allows simultaneous communication with the computer from up to eight remote Teletype<sup>2</sup> stations. The master controller consists of

two identical units, each of which controls up to four diffractometer stations. It decodes signals going to and coming from the station controllers, and handles certain functions which are common to all stations. These

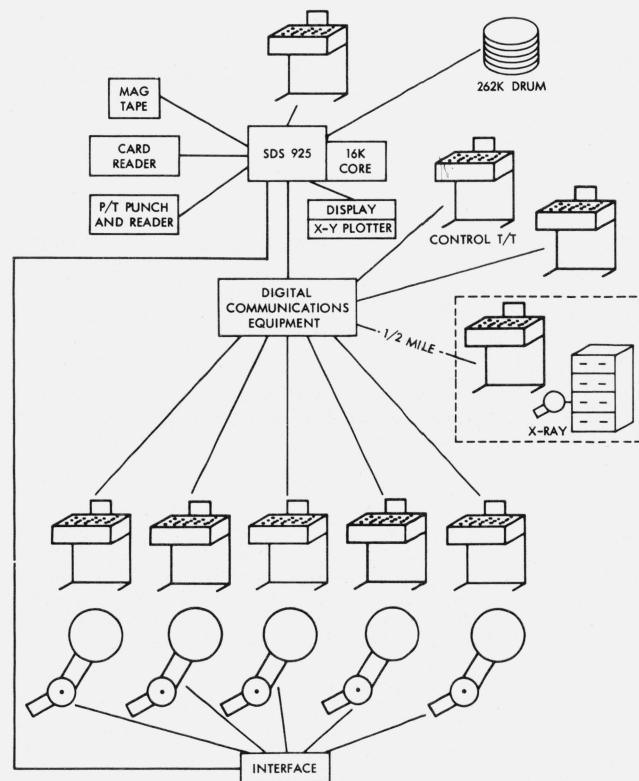


FIGURE 1. *Basic hardware of diffractometer control system showing 5 neutron and 1 x-ray diffractometers interfaced to computer. Each station has its own teletype unit.*

include a pair of digital to analog converters for data display, a 7.5-Hz clock function, light pen detector circuits, and an adjustable timer used for controlling motors. The station controllers contain equipment which is specific to a particular diffractometer. This equipment includes a 16-bit scaler for counting the diffracted beam, a 20-bit scaler for the incident beam monitor, six circuits for detecting "degree marks" which are returned once per revolution of a motor, six circuits for detecting if any motor has hit a limit, five circuits to detect the state of breakpoint switches, and six relay drivers.

## 3. System Programs

The necessary computer programs may be arbitrarily divided into system programs, which control over all operations, and user programs, which apply to computation and control for a user's particular experiment. System programs are divided into an "offline" (non-real-time) system and an "online" (real-time) system. The offline system described in detail below contains programs needed for general housekeeping.

<sup>1</sup> Xerox Data Systems, Inc., El Segundo, California. Certain commercial equipment is identified in this paper in order to describe adequately the hardware configuration of this system. In no case does such identification imply recommendation or endorsement by the U.S. Government, nor does it imply that the equipment identified is necessarily the best available for the purpose.

<sup>2</sup> Teletype Corp., Skokie, Illinois.

These include the FORTRAN compiler, loader, and library; the assembly routine; programs for such utility functions as copying paper tapes and initializing the system; a program for updating the system; and two programs for transferring control to the online system, one to be used after an orderly shutdown, and another to be used after an accidental or unplanned shutdown.

The online system contains the programs needed to control the experiments and for day-to-day routine operations. It is divided into a core-resident part, and other parts which reside on the disk, to be loaded into core when needed. The core-resident system includes the time-sharing executive routine, referred to as the distributor, subroutines for servicing priority interrupts, all routines required by the FORTRAN runtime package, and a set of subroutines which are used very frequently by users' programs, including all programs for control of diffractometer operations, as well as frequently used computation routines such as SIN, COS, ATAN, and EXP.

To each station there is assigned a specific 72-word block of core memory. Forty of these words are devoted to buffers, 20 words each for input and output, for temporary storage of Teletype messages. The remaining 32 words are pointers, counters, and flag words. One particular flag word, known as the status word, contains bits which are 1's when certain real-time operations (such as the movement of a motor or the counting of neutrons) are in progress, and 0's otherwise. Most of the other flag words are usually equal to -1, but are set to 0 when a real time operation is completed. The distributor repeatedly tests these flag words, and when one is equal to 0 a branch is entered which takes appropriate action. This usually means setting a bit in the status word to 0.

Action is initiated by typing a control message, which is a dollar sign followed by a four- to eight-character code word, and up to seven parameters. A carriage return or an exclamation point terminates the control message, and the control-message-finished flag is set. The distributor will then cause the control message processor to be read into core from the disk, and control is transferred to it. The control message processor is a disk-resident program which interprets control messages entered on a Teletype and initiates appropriate action. In addition to initializing the execution of programs, other actions include the output of information regarding the status of the system and control of the disk files that may be used by the experimenter. When the control message has been processed, control is returned to the distributor, which resumes testing.

After testing all flags, the distributor tests all status words. If the station is active, but the status word indicates that no real time operation is in progress, a user program will be entered. First, the station's COMMON area (i.e., locations where variables listed in a COMMON statement in a FORTRAN program are stored) is read into core from the disk. Attached to the COMMON area are several words which enable the

distributor to load the appropriate user program into core from the disk, and then to transfer control to the proper point in the program. The program may retain control for up to five seconds—a FORTRAN function is available to determine if time is almost up—and then must yield control. The COMMON area is now written back on the disk, and control is returned to the distributor. Thus the user saves needed variables (whose values would otherwise be lost upon yielding control to another station) by simply designating them to be in COMMON.

## 4. User Programs

User programs are two classes, designated level 1 and level 2. A level 2 program is restricted to 128 words of COMMON. In return for this restriction it will be given priority handling by the distributor, which will handle it before any level 1 program. Level 2 programs are intended for operations, such as "manual" operation of the diffractometers, during which a human operator is actually present waiting for results. A level 1 main program (maximum length 5508 words) may call up to a maximum of six segments (each with a maximum length of 2548 words) that will overlay one another in core. There may be up to 1024 words of COMMON. Thus the system can accommodate programs of great complexity, provided that they can be broken up into segments of less than 5 seconds of computing time each.

One special relationship between level 1 and level 2 programs is provided by use of the control message \$PAUSE, which causes the level 1 program to halt temporarily. While in this status, a level 2 program, such as one that enables "manual operations" to check on crystal alignment or spectrometer performance, may be executed. Finally, return to the original level 1 program is accomplished with \$RESUME.

## 5. Control of Real-time Operations

Real-time operations (such as moving a motor or starting the neutron counter) are controlled by means of a set of special subroutines that may be called by the user's FORTRAN program. The user's program computes the necessary parameters, calls the subroutines which initiate the required operation, and then yields. The subroutines set the appropriate bits in the station's status word, so that the station remains in a yield status until the real-time operation is completed. When the distributor has reset all of these bits, control then returns to the next statement after the yield. The following sections describe each real-time operation in detail.

### 5.1. Teletypes

Each station has its own Teletype, which communicates with the computer through an interface which is independent of the computer's main input-output

buffer. The Teletypes are wired in the full duplex mode, so that input from the keyboard and output to the printer are independent. On input the computer identifies the sending station, examines the character to see if it has a special significance, and if it is valid in context, converts the code to its internal code, stores the character in a 20-word buffer in the station's reserved area in the core, and acknowledges receipt of the character by retransmitting the character to the same Teletype's printer. A user program may request input by calling a subroutine TYPEIN, which sets the input-expected bit in the station's status word, and then automatically yields. An exclamation point or a carriage return terminates the message, after which the program may read the message in the buffer by means of the standard FORTRAN ACCEPT statement. If the program has not yielded at a TYPEIN, only control messages (those preceded by \$) may be entered. On output, characters are stored in a 20-word output buffer. The program then yields and the computer transmits the characters under interrupt control.

An x-ray diffractometer, located in a building remote from the computer, contains its own control hardware, and thus requires only Teletype communication to operate automatically. The messages sent by the computer are interpreted as instructions by the diffractometer, which sends back data to the computer.

The subroutine TYPEIN (whereby the user program calls for input) also starts the paper tape reader on the Teletype so that if a tape is in position it will be read. When a station becomes inactive (usually because a program has finished execution) the distributor will start the paper tape reader at that station. This feature is very useful because it makes it possible to execute a sequence of programs.

## 5.2. Neutron counting

Each neutron diffractometer has two pulse-counting channels, one for a monitor detector and one for the diffracted beam, connected to scalers built into the computer interface. The output of the diffracted-beam channel is also connected to a count-rate-meter which drives a strip chart recorder,<sup>3</sup> which provides a continuous time record of the detector output that is useful for monitoring the overall performance of the system during data collection. Either scaler may count either neutrons or 60-Hz timing pulses. The program calls one of the subroutines COUNT(MON) or CNTIME(MON) to count for MON monitor counts from detector or 60-Hz respectively, and then yields. Both scalers are started by a signal from the computer, and count until the monitor scaler overflows. Counting then stops, and the subroutine SCALER(ICOUNT) is used to read the diffracted beam count into location ICOUNT.

## 5.3. Motor Control

All neutron diffractometer angles are measured by means of worm gears or helical gears and worms driven by "Slo-Syn" stepping motors.<sup>4</sup> To start a motor the

program calls a subroutine START(MONUM, TARG) for the fast, slewing speed, or STARTS(MONUM, TARG) for the slow, scanning speed. MONUM is the motor identification number, and TARG is the angle to be reached. The program starts all motors which must be moved and then yields.

Motors may be assigned to any of eight frequencies. The stepping motors run more reliably at fast speeds if they are started by increasing the frequency in several steps. The system makes provision for starting motors in three steps, with frequency ratios of about 1.4. Also, to take care of backlash in the gears, the system causes all motors which are driven in a negative direction to overshoot the target by 0.2 degree, and then approach the final angle in the positive direction.

If the angle-measuring device driven by a motor hits a limit switch, it is an indication either of a programming error or of a malfunction which has caused the true angle to be different from the angle recorded in the computer. In either case the system program stops the motor, makes the station inactive, and prints a message on the station's Teletype.

## 5.4. Degree Markers

Experience has shown that, if motors are properly installed and have been assigned to appropriate frequencies, slippage is a rare occurrence, but it must be detected if it does occur. For this purpose each motor is fitted with a degree marker, which consists of a lamp, a disk with a slot in it attached to the motor shaft, and a photodiode detector. The assembly is adjusted so that the slot will allow light to hit the photodiode during the step immediately before a whole degree. If an error is detected a correction is made. If the number of errors exceeds a predetermined number, the motor is flagged as malfunctioning. The motor is stopped, and the station is made inactive.

## 5.5. System Clock

The system clock keeps track of the time of day and the day of the year, and performs two other system functions. It determines whether a station has used the computer for more than 5 seconds at a time, and stops the program if the time limit is exceeded. It also allows the programmer to call a subroutine HOLD (TIME) which automatically yields for a specified time interval and then resumes computing.

## 5.6. Breakpoint Switches and Relays

The computer interface provides five points per station which can be tested by the computer, using the FORTRAN statement IF (SENSE SWITCH N) to control branches in the program. These points are connected to toggle switches on the typewriter panel, and may also be connected to switches controlled by real-time operations, such as a temperature indicator or a position indicator. Each station is also provided with drivers for five relays, which can be set or reset by means of program instructions. These functions may be combined to perform such operations as measuring intensity versus temperature or changing filters.

<sup>3</sup> Rustrak Instrument Co., Manchester, N.H., Model 88/109B.  
<sup>4</sup> Manufactured by Superior Electric Co., Bristol, Conn.

### 5.7. Data Display

The computer interface provides two analog outputs, which can be set to values in the range  $\pm 10$  V by means of program instructions. These outputs are fed into the  $X$  and  $Y$  inputs of a storage oscilloscope<sup>5</sup> and an  $X-Y$  plotter.<sup>6</sup> A FORTRAN subroutine, CONVRT (IX, IY) sets the outputs to levels proportional to the values of IX and IY in the range 0 to 511. Another subroutine, DISPLAY (IYO, ISIZE), if called as the next executable statement after a TYPE statement, will cancel the typing and instead will display the characters of the output record on the oscilloscope screen, enabling the programmer to label a data display with identifying information. The arguments specify the height on the screen where the line of characters will be displayed and the size of the characters.

The experimenter may manipulate the data displayed on the oscilloscope by means of a light pen.<sup>7</sup> A word in core is incremented whenever the pen is in position to be activated by a spot flashed onto the screen. The value of the word is then available as the function LITEPN (N) which is tested by the program immediately after each particular spot has been displayed. The word is automatically reset to -1 each time the function is called.

### 5.8. Sequential Files on the Disk

151,552 words of disk memory, slightly more than one half of the total disk memory, are allocated for storage by the users of data and intermediate results of computations. This space is divided into files, which, from the programmer's point of view, simulate binary magnetic tape units. Files are divided into 128-word records, and are of two types: data files, containing 88 records; and scratch files, with eight records. The FORTRAN statements READ TAPE N and WRITE TAPE N will read information from and record information on file N. Two subroutines, RWNDSF(N) and SKPREC(N, ISKP) will, respectively, "rewind" the file N and "skip forward" ISKP records in the file. Control messages are used to reserve up to five files at each station, gain access to a file at another station (read only), or to transfer a file from one station to another. Logical unit numbers may be assigned to files and they may be write-protected if desired.

### 5.9. Other Input-Output

The computer has a magnetic tape drive unit, a card reader, and paper tape reader and punch which may be used by programs called from a Teletype station located in the computer room. The BCD tape operation statements of FORTRAN, READ INPUT TAPE N and WRITE OUTPUT TAPE N, may be used by experimenters for reading and writing magnetic tapes for permanently storing data or for communicating with other machines.

<sup>5</sup> Tektronix, Inc., Portland, Oregon, Model RM564.

<sup>6</sup> Hewlett-Packard, Pasadena, California, Moseley 7035B.

<sup>7</sup> Sanders Associates, Inc., Nashua, N.H., Photopen EOPT.

### 6. The Offline System

When the control system was still in the planning stage, it was concluded that it was not feasible to compile and load FORTRAN programs while sharing time with other operations. This would have required modifying the FORTRAN compiler itself to be a real-time program, a task which would have been far too difficult to justify in terms of the benefits that would have resulted. Therefore compiling and loading of programs, along with certain other less frequent operations, are done in a non-real-time, batch processing mode under the "offline system." Care was taken to make the transition from the online to the offline mode (and vice versa) automatic and foolproof; no data or motor angles are lost. The real-time operations are merely suspended upon going offline.

The compiler is a REAL TIME FORTRAN II, which was supplied by XDS and has not been changed in any way. Virtually all user programs are written in FORTRAN language and compiled using this compiler. The only exceptions are certain very short, special-purpose subroutines which perform operations outside of the restrictions of the FORTRAN language, and therefore must be written in assembly language.

In actual operation a source program, on punched cards, is compiled, producing a binary object program on punched paper tape. The program is assigned a number, by which it may be called during real-time operation, and is loaded into core. Library subroutines and links to core-resident subroutines are attached, in absolute address form. When loading is complete, the program and all attached subroutines are written on a program file on the disk, and the program's number is inserted in the program directory, an area on the disk which contains a catalog of the locations of all programs in the program area. The directory will then be scanned by the control message processor to determine the locations from which the program is to be loaded into core at run time.

The offline system also contains various programs for initializing the system, and for certain other utility operations.

### 7. User Software

The actual use of a computer may change one's approach to the problem at hand, but the implementation of revised ideas is very much a function of the amount of labor involved. The use of FORTRAN has greatly facilitated the development of programs by individual users. The extensive I/O capabilities of FORTRAN and the use of the COMMON statement to designate the locations of necessary variables have enabled application of some generally useful ideas that have emerged in the course of writing user programs. This section lists some of these ideas, and gives examples of how they have been used so far. The programs mentioned do not represent an inclusive list of the programs that have been written. The program library is still being steadily expanded.

### 7.1. Routine Tasks

Short, routine tasks frequently repeated require highly simplified input from the Teletype for maximum convenience. One example is a program that carries out simple tasks such as driving motors to a designated angle, counting in place, rocking a crystal through a given angular range, finding the monitor count, finding the location of a peak within a given range, etc. The operation code consists of two letters, such as "FM" for find monitor count rate and "CT" for count.

### 7.2. Experimental Parameters

Parameters that are used repeatedly in an experiment are entered once, and then saved in COMMON. For example, in the program described above the monitor count is entered with M=(18). Subsequently, every time "CT" is called, the same monitor count is used.

A more sophisticated example of this same idea is given by a separate program (called PRAM) which accepts all pertinent data for a single crystal, including cell constants, Laue point group, systematic absences, the orientation of the crystal, etc. This program stores these quantities in COMMON, along with certain derived quantities such as the orientation matrix, where they are available to other programs that may now easily compute the setting angles for any reflection,  $hkl$ , or use other information supplied by PRAM.

### 7.3. Data Handling

The need for the ability to reduce and manipulate data with a rapid turnaround varies widely from one experiment to another. In some cases it may be sufficient to write data in "raw" form on a magnetic tape, leaving all further processing to a central computing facility. In other cases it may be desirable for the user to exercise human judgement in the course of an experiment. Judgement may be shared between the user and the computer in varying proportions depending on the circumstances. It is important to have available, in addition to the actual online control programs, data-handling programs sufficient to perform whatever manipulations may be required to provide the information that is needed quickly.

As an illustration of a rather complete capacity for handling data, consider the set of programs written for processing powder data (i.e., neutron intensity as a function of counter angle). These data are taken by one program which stores them with suitable header information, which includes a problem number and run number, and five integer parameters which the experimenter may use to label conditions of the run, such as the temperature, magnetic field, etc. This program collects the data, stores them on a data file, and prints out the information as it is accumulated, if desired. This record is not very compact since each line contains only one intensity point. This program may be used repeatedly to collect many runs, which

may be repeats or runs under different experimental conditions. A second independent display program will read the data and print it out in compact form, display it on the storage scope, or plot it on the  $x-y$  recorder. Scaling may be done automatically by the program or left to the choice of the user. Background points may be indicated by the user by means of the light pen, or on the Teletype, and commands given for the net integrated intensity of a peak to be determined, together with its statistical error. A third program enables the experimenter to collate any data runs by adding or subtracting two or more of them. The results of the collation are also stored on a file and may be subsequently processed by the second program. Printouts of collated runs contain the statistical errors for each data point. There are two additional programs, one for writing data files from the disk onto magnetic tape and the other for recalling data from magnetic tape back onto the disk where it can be reprocessed. The recall program utilized problem and run numbers and the five integer parameters mentioned above to search the tape for the desired data.

The system feature which allows one station to have access (read-only) to another station's files allows all the manipulative and analysis functions described above to be carried out while data collection is still going on at the experimental station.

### 7.4. Modification of Programs

It would appear to be an axiom of computer usage that the full specifications for a complicated program or system of programs are never completely known beforehand, but rather evolve during the course of writing, testing, and using the programs. A typical example is illustrated by the powder data program mentioned above, which, because of the FORTRAN capability, was easily modified to collect data as a function of  $\sin \theta/\lambda$  ( $\theta$  = counter angle,  $\lambda$  = wavelength) where amorphous materials were measured. A further modification enables one to collect data for a single crystal for scans along some specified direction in reciprocal space. In all three cases the data finally stored on the data file are in the form of intensity as a function of some continuous variable. Thus, the manipulative and analysis program described in the previous section could be used as written.

### 7.5. Automation of Routine Tasks

Certain routine tasks, if automated, will be used more frequently with better overall results for the experiment.

One example of such a task is the measurement of the neutron wavelength and the zero correction to the counter angle. A copper powder sample is used as a standard and four reflections are scanned for their angular positions. The only input required to the program is an approximate value of the wavelength. The program determines the best, least squares value of the wavelength and zero correction.

## 7.6. Feedback

Utilization of feedback enables great efficiencies to be realized in data taking. For example, when measuring single crystals with moderately large unit cells, for which many hundreds of reflections are required, large amounts of time may be wasted by scanning reflections which are so weak that they are essentially unmeasurable or by measuring strong reflections to an accuracy greater than is needed. A program was therefore written which performs a quick scan over the peak to estimate the intensity first and then proceeds to measure the reflection to the desired accuracy. This program also checks several successive measurements of the intensity for statistical consistency, and causes the data to be remeasured if they are not consistent.

## 8. Operating Experience

In 22 months of operation the system has proved itself to be highly flexible and reliable. Because the designers and system programmers are also user-scientists, this has permitted modifications to the system program to be made easily when the need was apparent. More than 10 different people have written programs for use on the system with a minimum of difficulty. In some cases the fact that programs are written in FORTRAN has enabled the use of programs which were originally written for quite different machines under quite different circumstances, without any modification whatsoever. The FORTRAN compiler, with its diagnostics, is so effective in eliminating user programming errors that in practice it has been found that all inhibitions to writing new programs for fear they might adversely affect other user programs have been removed. Only two ways have been found for a programming error to cause a system malfunction: writing in an array with an incorrectly computed value for a subscript, and use of an incorrect name for a subroutine. The latter error can cause trouble only if the incorrect name also appears in the library with a special significance.

In the unlikely event that the system does crash, it may be restarted with a single control message and the

only variables that must be restored at each station are the motor angles. The user program may test for a restart condition and take special action if that is needed; otherwise the user program will automatically continue from the statement after which it last yielded.

The stepping motors have proved to be highly reliable, provided they are operated within their ratings with respect to speed and torque. When properly adjusted they have been shown to operate for tens of thousands of revolutions without a single slippage error. In a few cases it has been necessary to experiment with a motor to find its proper speed, but the fact that speeds are determined by programs means that they can be changed by simple program changes, often no more than the change of one or two words in a system program.

Because different users can communicate with the computer simultaneously through the individual station Teletypes, instances of interference between users for access to input-output facilities have been extremely rare. The system has proved, so far, to satisfy its design objectives extremely well.

---

The basic concepts of the time sharing and the FORTRAN capability were developed jointly with Gerd Dimmler of the Brookhaven National Laboratory. His valuable contributions are gratefully acknowledged. We also thank Louis Paoellla, Alan Tudgay, and Robert Williams for their assistance in installing the system hardware.

## 9. References

- Abrahams, S. C. (1962), Rev. Sci. Instr. **33**, 973.  
Beaucage, D. R., Kelly, M. A., Ophir, D., Rankowitz, S., Spinrad, R. J., and Van Norton, R. (1966), Nucl. Instr. and Methods **40**, 26.  
Busing, W. R., Ellison, R. D., Levy, H. A., King, S. P., and Roseberry, R. T. (1968). Report ORNL-4143, Oak Ridge National Laboratory, Oak Ridge, Tenn.  
Fitzwater, D. R. (1965). Report IS-1279, Ames Laboratory, Iowa State University.  
Kruger, G., and Dimmler, G. (1964). Report KFK-242, Kernforschungszentrum Karlsruhe, Germany.  
Prince, E., and Abrahams, S. C. (1959), Rev. Sci. Instr. **30**, 581.

(Paper 74C3 & 4-302)