IS-4378

MASTER

PROGRAMMERS GUIDE TO FINGER, THUMB, AND TRIPLE AXIS

REAL TIME EXPERIMENT CONTROL PROGRAMS

Thomas G. Pinter
and
Elaine M. Notis

AMES LABORATORY, DOE
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: March 1978

PREPARED FOR THE DEPARTMENT OF ENERGY
UNDER CONTRACT W-7405-eng-82

IS-4378

PROGRAMMERS GUIDE TO FINGER, THUMB, AND TRIPLE AXIS REAL TIME EXPERIMENT CONTROL PROGRAMS

Thomas G. Pinter and Elaine M. Notis

Ames Laboratory, USDOE
Iowa State University
Ames, Iowa 50011

Date Transmitted: March 1978

PREPARED FOR THE U. S. DEPARTMENT OF ENERGY, DIVISION OF BASIC ENERGY SCIENCES UNDER CONTRACT NO. W-7405-eng-82

NOTICE

Printed in U.S.A.

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche $3.00
Paper Copy $15.00

TABLE OF CONTENTS

ABSTRACT

This manual presents the algorithms and various programming details of the PDP-15 and SDS-910 codes for real time experiment control of the neutron scattering spectrometers. The mathematical basis for the codes as well as the data set organization are also discussed.

Distribution List

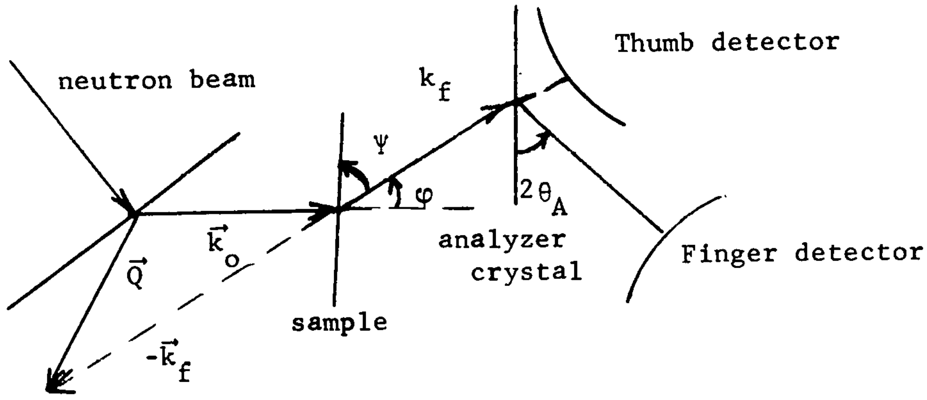
Chicago Patent Office	1
USDOE - TIC	27
Ames Laboratory Library	5
Thomas G. Pinter	6
Elaine M. Notis	1
Total	40

I. INTRODUCTION

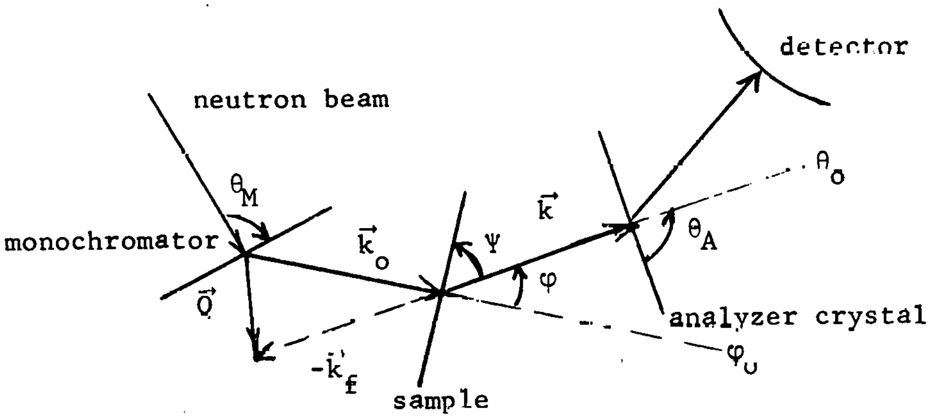
This manual describes the tasks written to control the neutron scattering real-time experiments. These experiments consist of three diffractometers, called Thumb, Finger, and Triple Axis, as well as an x-y Plotter. The basic control system consists of an experiment interfaced to an SDS-910 which is in turn interfaced to a PDP-15. All direct experiment control is performed in the 910, e.g., setting angles, handling experiment interrupts, etc. Thus, all three experiments "look" the same to the 15.

The Thumb diffractometer consists of a sample table of rotation for the crystal being studied and a detector arm which is rotated about the sample table axis. All other angles are fixed. Thus, only elastic scattering can be performed on this diffractometer. The Finger diffractometer has an analyzing crystal where the Thumb detector is located. This crystal can be rotated to select a given energy from the beam scattered by the sample before it enters the detector. Thus, inelastic scattering can be performed. See Figure 1.a for a diagram of the Finger and Thumb diffractometers. The Triple Axis resembles the Finger except the monochromator and drum angles as well as the detector and analyzer crystal angles can all move independently. See Figure 1.b for a diagram of the Triple Axis diffractometer.

On the Finger and Thumb diffractometers, the sample table and detector arm are coupled. This means that when the sample table rotates through an angle φ, the detector moves through an angle $\varphi/2$.



a. Mitsubishi



b. Triple Axis

Figure 1. Experimental layout for Mitsubishi and Triple Axis diffractometers.

In order to perform an uncoupled scan, the detector must be moved through an angle $-\varphi/2$ after the sample table has been moved through an angle φ. This detector motion is not shown on the angle readout display. Thus the arms appear to move for uncoupled scans, and appear not to move for coupled scans.

The Thumb and Finger experiments are each controlled by a Mitsubishi diffractometer with its own internal processing unit. This unit drives the tables and detector arm as well as controls the count. These diffractometers can be driven manually from a console on the diffractometer as well as automatically from a paper tape reader. The same interface that connects the paper tape reader to the Mitsubishi serves to handle the 910 control as well. Thus, the 910 must emulate a paper tape when sending scans for the Mitsubishi to process. The Triple Axis diffractometer, on the other hand, has no internal processing unit. Therefore, the 910 exercises direct control of this diffractometer. It must monitor the arm motion, send start commands for counting etc. directly. Hence, although the 15 codes are similar for these three diffractometers, the 910 codes are quite different.

Each experiment has three scans which can be performed: a general step scan, a double rock, and a q-e scan. Each of these scans will be discussed separately. The general step scan is a series of tasks which drives a given angle or angle pair through a set of equally spaced angles, counting for a given length of time at each angle setting. The double rock scan is a set of two general scans performed on a given angle pair. After each separate scan, the weighted mean is calculated. This mean becomes the center angle setting for the completed scan during the next scan of the pair. These pairs of scans are repeated until the change in each mean for two consecutive scans is less than some determined value.

The q-e scan is a scan in reciprocal space through a set of equally spaced momenta and/or energies. In order to generate this scan, the momenta and energy in reciprocal space must be transformed to a set of angles in real space. This transformation and the weighted mean calculation will be described in the following section.

II. MATHEMATICAL METHOD

A. Angle Calculation

In the following section the equations used to calculate the various angles used in q-e scans on the Mitsubishi and Triple Axis diffractometers shall be described. Certain definitions and relations will be helpful throughout this description. They are given below.

λ	- wavelength of neutron radiation in Angstroms.
λ_o	- incident wavelength.
λ_f	- final wavelength.
$k=2p/\lambda$	- magnitude of the wavevector. The direction of $\vec{k}$ is the real space direction of the neutron beam.
$E=\hbar^2k^2/2m_n$	- the energy of a neutron of wavevector $\vec{k}$ where $\hbar$ is the Planck constant divided by 2π and m_n is the mass of the neutron. These energies are typically on the order of .025eV.

$\vec{p}=\hbar\vec{k}$ - the momentum vector. By the relationship $\vec{p}=\hbar\vec{Q}$, the scattering vector determines the momentum change of the neutron upon scattering, and thus the momentum transferred to the sample.

$\vec{G}_1,\vec{G}_2$ - the two reciprocal lattice vectors used to calibrate the instrument and sample. These two vectors may not be colinear. Any position in reciprocal space may be defined in terms of these two vectors.

φ - the angle between $\vec{k}_o$ and $-\vec{k}_f$. The common instrumental names associated with φ are GA and $2\theta_{S1}$, which both involve the addition of some zero reference. For example,

$$GA = \varphi + GA_o$$

whereas φ is an absolute angle determined only by geometry and not instrument parameters.

δ - the angle from the calibration vector $\vec{G}_1$ to the scattering vector $\vec{Q}$.

γ - the angle from the scattering vector $\vec{Q}$ to the incident wavevector $\vec{k}_o$.

ϵ - a reference angle stating the instrumental position of the sample table less γ when the instrument is set to measure $\vec{G}_1$. ϵ is arbitrary depending entirely on the instrument.

η - the angle between the calibration reciprocal vectors. η rotates $\vec{G}_1$ into $\vec{G}_2$ in a right hand sense.

Conservation Relations on Scattering

momentum - $\vec{Q}=\vec{k}_o-\vec{G}+\vec{q}$ where $\vec{G}$ is a reciprocal lattice vector and $\vec{q}$ is the wavevector of some excitation in the lattice.

energy - $E_f-E_o=(\hbar^2/2m)(k_f^2-k_o^2)=h\nu$ where $h\nu$ is the energy of an excitation in the lattice. For elastic scattering, $h\nu=0$. For the creation of an excitation, $h\nu<0$. For the annihilation of an excitation, $h\nu>0$.

The calibration experiments relate the geometry of the diffractometer with its orientation in momentum space. The calibration measures Ψ_1, Ψ_2, φ_1, and φ_2 (see Fig. 2).

$$\vec{Q}=a\vec{G}_1+b\vec{G}_2$$
$$\text{where } |\vec{G}_1|=2|\vec{k}_o|\sin(\varphi_1/2)$$
$$|\vec{G}_2|=2|\vec{k}_o|\sin(\varphi_2/2) \qquad (1)$$

since the momenta form an isosceles triangle if the scattering is elastic. If the unit vectors are denoted as $\vec{e}_i$, then

$$Q_1\vec{e}_1+Q_2\vec{e}_2+Q_3\vec{e}_3=(ah_1+bh_2)\vec{e}_1+(ak_1+bk_2)\vec{e}_2+(a\ell_1+b\ell_2)\vec{e}_3$$
$$\text{or } Q_1=ah_1+bh_2$$
$$Q_2=ak_1+bk_2$$
$$Q_3=a\ell_1+b\ell_2 \;. \qquad (2)$$

Solving equations (2) for a and b yields the forms:

$$a=\frac{Q_1k_2-Q_2h_2}{h_1k_2-h_2k_1} \quad b=\frac{Q_2k_1-Q_1h_1}{h_1k_2-k_1h_2}$$

$$a=\frac{Q_1h_2-Q_3\ell_2}{h_2\ell_1-h_1\ell_2} \quad b=\frac{Q_3h_1-Q_1\ell_1}{h_3\ell_1-h_1\ell_2}$$

$$a=\frac{Q_2\ell_2-Q_3k_2}{k_1\ell_2-\ell_1k_2} \quad b=\frac{Q_3k_1-Q_1\ell_1}{k_1\ell_2-\ell_1k_2} \qquad (3)$$

Any of the three forms may be used. However, since the region of q-space employed in any one measurement is only two dimensional, some of the solutions for a and b will be indeterminate. Thus, care must be taken to select a non-zero denominator to calculate a and b.

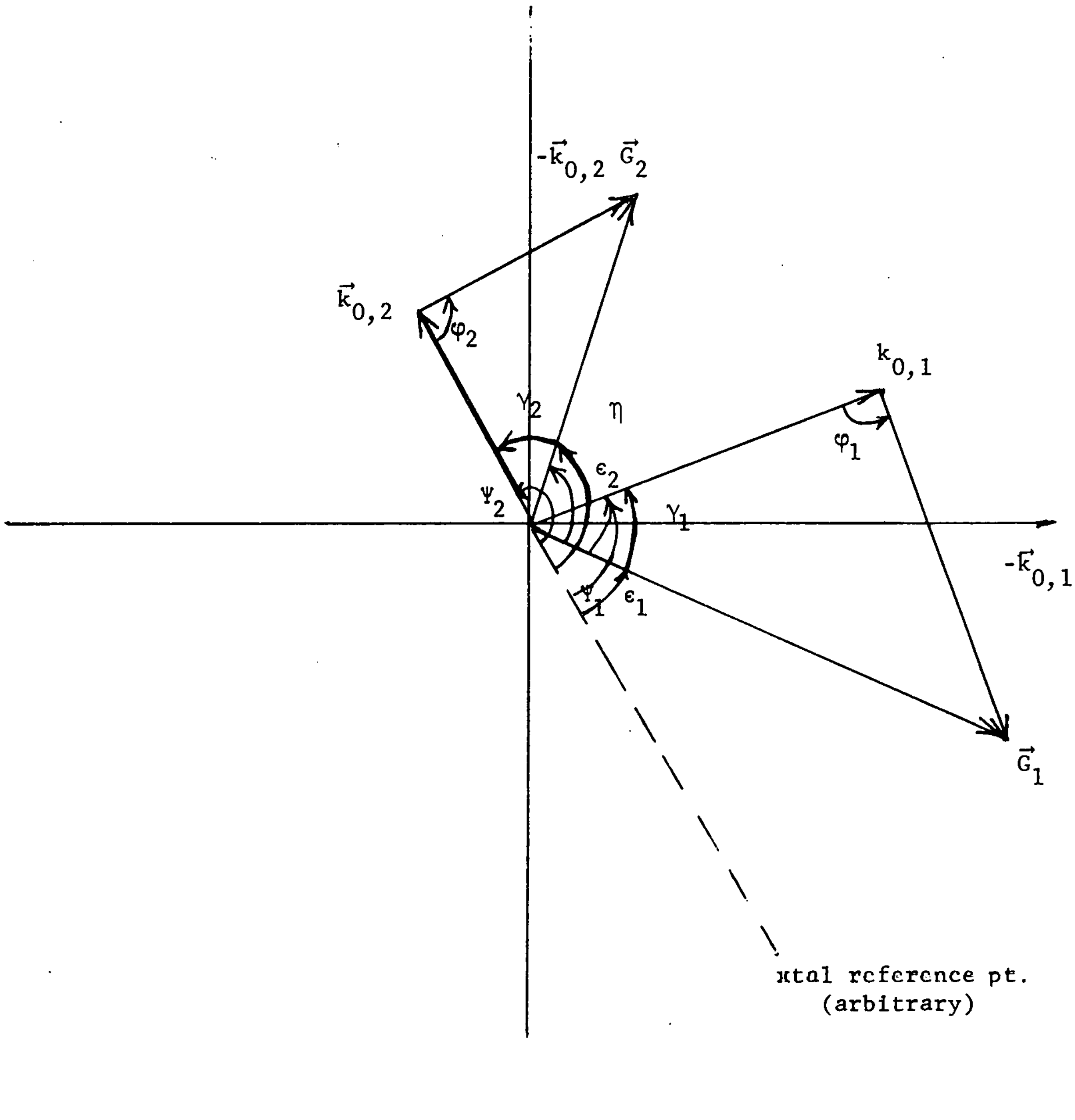


Figure 2. Relationship of calibration vectors and angles.

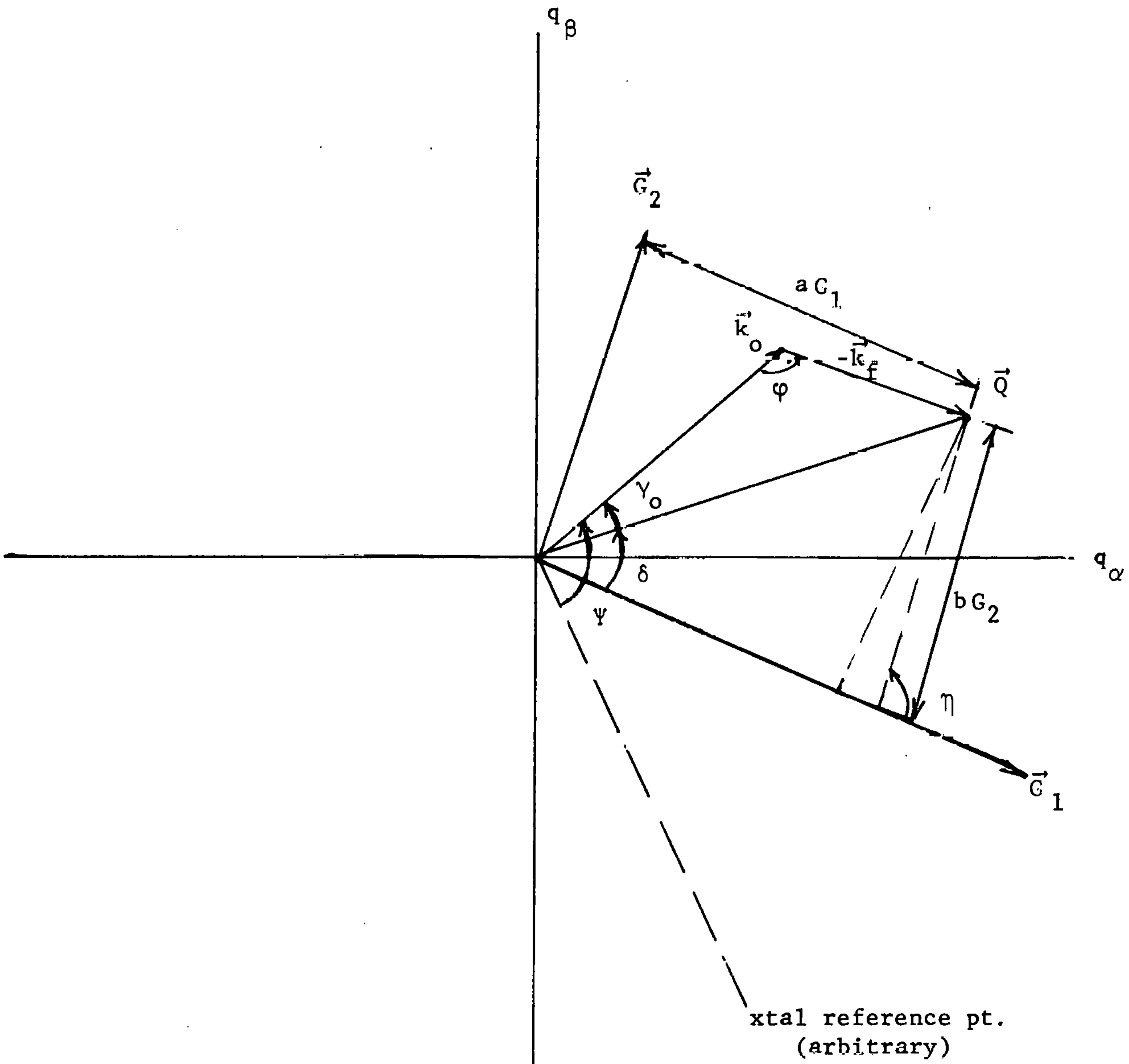


Figure 3. Relationship of calibrations and $\vec{Q}$ and angle relationships.

Now from Fig. 2, the following angle relationships can be written:

$$\Psi_i = \epsilon + \gamma_i$$
$$\gamma_i + \gamma_i + \varphi_i = \pi$$
$$\gamma_i = (\pi - \varphi_i)/2$$
$$\Psi_2 - \Psi_1 = \epsilon_2 + \gamma_2 - \epsilon_1 - \gamma_1$$
$$\Psi_2 - \Psi_1 = \epsilon_1 + \eta + \gamma_2 - \epsilon_1 - \gamma_1$$
$$\Psi_2 - \Psi_1 = \eta - (\pi - \varphi_1)/2 + (\pi - \varphi_2)/2 = \eta - (\varphi_1 - \varphi_2)/2$$
$$\text{or} \quad \eta = \Psi_2 - \Psi_1 + (\varphi_1 - \varphi_2)/2 \qquad (4)$$

From Fig. 3

$$(\vec{Q} - a\vec{G}_1) \cdot (\vec{Q} - a\vec{G}_1) = b\vec{G}_2 \cdot b\vec{G}_2$$

$$Q^2 + a^2 G_1^2 - 2aG_1 Q\cos\delta = b^2 G_2^2$$

$$\delta = \tan^{-1}(bG_2 \sin\eta / (aG_1 + bG_2 \cos\eta)) \qquad (5)$$

$$(a\vec{G}_1 + b\vec{G}_2) \cdot (a\vec{G}_1 + b\vec{G}_2) = \vec{Q} \cdot \vec{Q}$$

$$a^2 G_1^2 + 2 + b^2 G_2^2 + 2\cos(\pi - \eta) aG_1 bG_2 - Q^2$$

$$Q = \mathrm{sqrt}(a^2 G_1^2 + b^2 G_2^2 - 2\cos\eta\, G_1 G_2 ab) \,. \qquad (6)$$

Finally from Fig. 3

$$\Psi = \epsilon_1 + \delta + \gamma_0$$
$$= \Psi_1 - \gamma_1 + \delta + \gamma_0$$
$$= \Psi_1 - (\pi - \varphi_1)/2 + \delta + \gamma \qquad (7)$$

For elastic scattering $\gamma = (\pi - \varphi)/2$ and $\varphi = 2\sin^{-1}(Q/2k_o)$ from Eq. 1. Since only elastic scattering can be performed on the Thumb diffractometer, these equations are sufficient to calculate the necessary angle settings φ and Ψ. Inelastic scattering can be performed with the Finger and Triple Axis spectrometers. Since $k_f \neq k_i$ for inelastic scattering, the cosine law yields the relationship

$$\cos\varphi = (k_f^2+k_i^2 - Q^2)/2k_f k_i$$

$$\cos\gamma = (Q^2+k_i^2 - k_f^2)/2Qk_i$$

and the sin law yields the relationship

$$\sin\varphi = Q\sin\gamma_o/k_f .$$

Thus,

$$\gamma = \cos^{-1}((Q^2+k_i^2-k_f^2)/2Qk_i)$$

$$\text{and } \varphi = \tan^{-1}(2k_i Q\sin\gamma/(k_f^2+k_i^2-Q^2)) . \quad (8)$$

Hence φ and Ψ for inelastic scattering is determined by Eqs. 3, 4, 5, 6, and 8.

For the Thumb, these are all of the required angles. For the Finger and Triple Axis, the analyzer crystal orientation has to be calculated. This is done using the Bragg Law.

$$2d\sin\theta = \lambda \text{ or } \theta = \sin^{-1}(\lambda/2d) \quad (9)$$

For the Triple Axis diffractometer, the monochromator angle is also calculated from the Bragg Law.

Finally, several angles can rotate in either sense from their reference lines. Thus, the codes allow for both the positive and negative sense of rotation. Angles signs are positive for rotation in the right hand sense and negative for rotation in the left hand sense.

B. Area Calculation

The area calculation consists of three distinct steps: location of the background; removal of the background from the spectrum; and the calculation of the centroid or weighted mean of the spectrum. For these purposes the spectrum has six salient features: the start of the left background, the end of the left background, the start of the left side of the peak, and the respective features on the right side of the spectrum.

The background is determined as follows: the average standard deviation of the first three points is calculated. Next the average of the three points starting at point 1 is compared with the average of the three points starting at point 2. If the difference is greater than 1 standard deviation, then this is noted. Comparison continues for points 2 and 3. When three consecutive comparisons are greater than the standard deviation, the end of the background is marked at the first comparison greater than the average standard deviation. The peak is marked whenever the number of counts is greater than ninety per cent of the maximum number of counts in the spectrum.

Any features determined by the task are printed on the teletype. For the backgrounds, the average count and key position in the count file are printed. For the peak, the starting and ending key positions are printed.

If backgrounds are determined on both sides of the spectrum, then a straight line is fitted to these background points. This straight line is then subtracted from the spectrum. In this manner the spectrum can be corrected for sloping backgrounds. The equations of the fit are:

$$\text{slope}=b=\left(\sum_i i \sum_i y_i - n\sum_i y_i\, i\right)/D$$

$$\text{intercept}=a=\left(\sum_i i \sum_i i y_i - \sum_i i^2 \sum_i y_i\right)/D$$

$$\text{where } D=\left(\sum_i i\right)^2 - n\sum_i i^2 \qquad (10)$$

The sum over i contains the points in the left background and the right background only. n is the total number of points in the sum. The i refers to the location of each point in the spectral array. Conversion to proper units occurs later.

The expression ai-b is then subtracted from each y_i in the spectrum. This causes the background to be properly scattered about a zero baseline as well as correcting the peak for any slope in the background. Finally, the weighted mean and the area are calculated and printed on the teletype. The area is calculated twice: once by straight summation, and once by Simpson's rule.

$$\text{AREA}_1=\sum_i y_i$$

$$\text{AREA}_2=(1/3)\sum_i (y_{i-2}+4y_{i-2}+y_i) \qquad (11)$$

where ELB and ERB are the end of the left and right backgrounds respectively.

If the peak has a plateau, i.e., if the difference between the keys marking the peak is greater than 7; then the mean is calculated as follows: The average of the counts through the peak is calculated. Three standard deviations of this average are then subtracted from it. The entire peak is then searched until the first point on each side of the peak greater than this value is found. The mean is just the average of these positions:

$$M=d\varphi\,((CR+CL)/2-1)+\varphi_o \qquad (12)$$

where $d\varphi$ is the increment of the spectrum abscissa, and φ_o is the initial abscissa values of the spectrum. If the spectrum does not have a plateau, the weighted mean is calculated by the expression:

$$M=d\varphi \sum(i-1)y_i/\sum_i y_i+\varphi_o\ . \qquad (13)$$

III. THE PDP-15 CODES

In this section the PDP-15 real time control codes will be discussed. The controlling codes for each diffractometer are similar in logic structure. The basic difference between the Triple Axis diffractometer and the Mitsubishi diffractometers is the addition of a second detector on the Triple Axis diffractometer and the possibility of many more scans because of the nine independent angles. Since the logic is similar, the codes will be discussed as a group with the diffractometer differences noted. Any terms that appear specifically in the codes will be printed in capital letters.

Since the computer is core bound rather than time bound, the ALECS language is a subset of PL/I which emphasizes tasking. This means that the programs are broken down into tasks which are loaded into core when called and removed from core when they are exited. This is an effective overlay structure which minimizes core usage for a given set of tasks. Thus, all variables have their compile time values each time the task is called. All variables not explicitly initialized are set to zero or blank.

In a tasking system, tasks can be executed asynchronously, i.e., several tasks are executed simultaneously in the same time frame, as well as synchronously, i.e., the execution of the calling task is suspended until control is returned from the called task. The asynchronous task call allows several operations to be performed concurrently, e.g., writing data to disk and printing on the teletype. The reader should refer to the PL/I language reference manual for a more detailed discussion of tasking*.

Since bit manipulation is common in these tasks, rotate accumulator functions have been implemented in the language. These are Rotate Accumulator Left (RAL) and Rotate Accumulator Right (RAR).

---

*IBM System/360 Operating System PL/I (F) Language Reference Manual File No. S360-29.

These allow information to be packed into one word without using fixed multiply and divide operations. The bit variables are declared fixed throughout as there is no fixed to bit conversion. Thus bit arithmetic is constantly done with fixed numbers. Therefore, operations with Boolean operators should be considered bit operations*.

The remainder of this section will be used to describe the individual tasks of the various task trees (see Figs. 4/5). The section will be divided into subsections to discuss the experiment codes, the plotter codes, and the area calculation and tape punching codes.

A. The Diffractometer Control Codes

The basic functions of any of the experimental codes is to set up the scan to be executed, collect the data, write the data to disk, and print the data to teletype. Since the computer or the reactor can misfunction, allowances are made for restarting the tasks at their point of interruption. This means that the data set containing the scan files must be updated as each data point has completed.

The teletypes automatically turn on whenever a character is sent to be printed. This is to allow the teletype motor to be turned off during the 90% of the time data are being collected. Since the teletype takes several milleseconds to get up to speed,

---

*If any bit in a variable is set, the variable is considered set. If no bits are set, then the variable is considered reset. A bit is set when it has the value 1 and reset when it has the value 0.

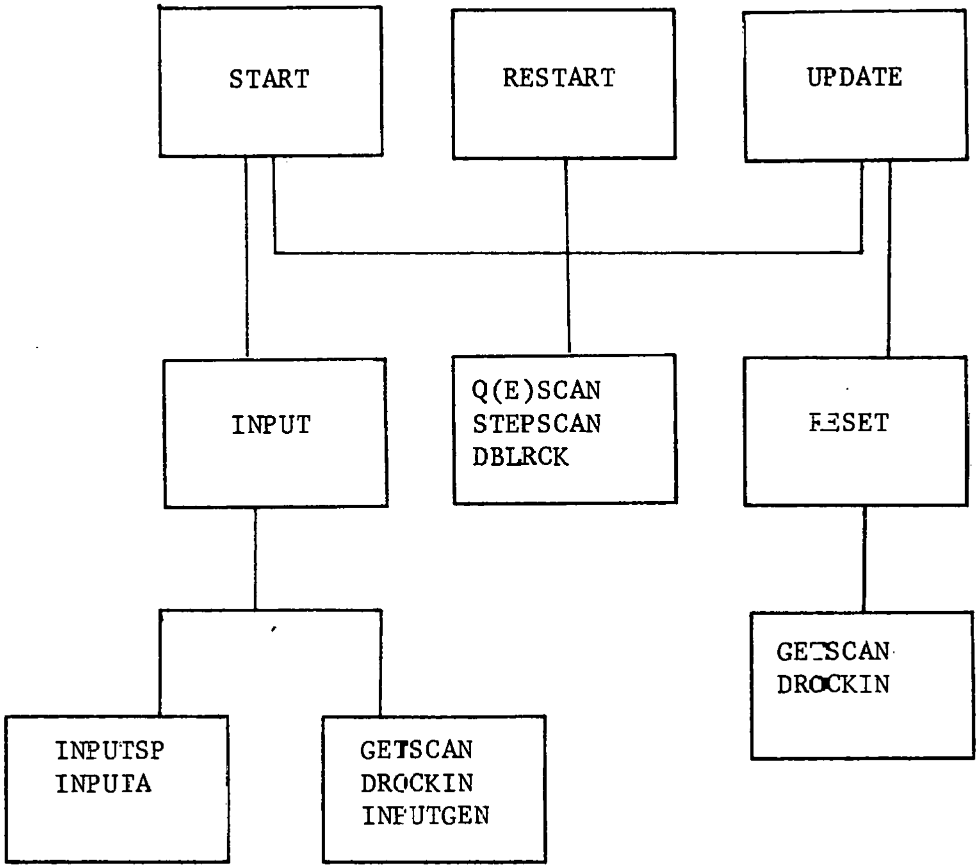


Figure 4. Task trees for initiating real time control experiment.

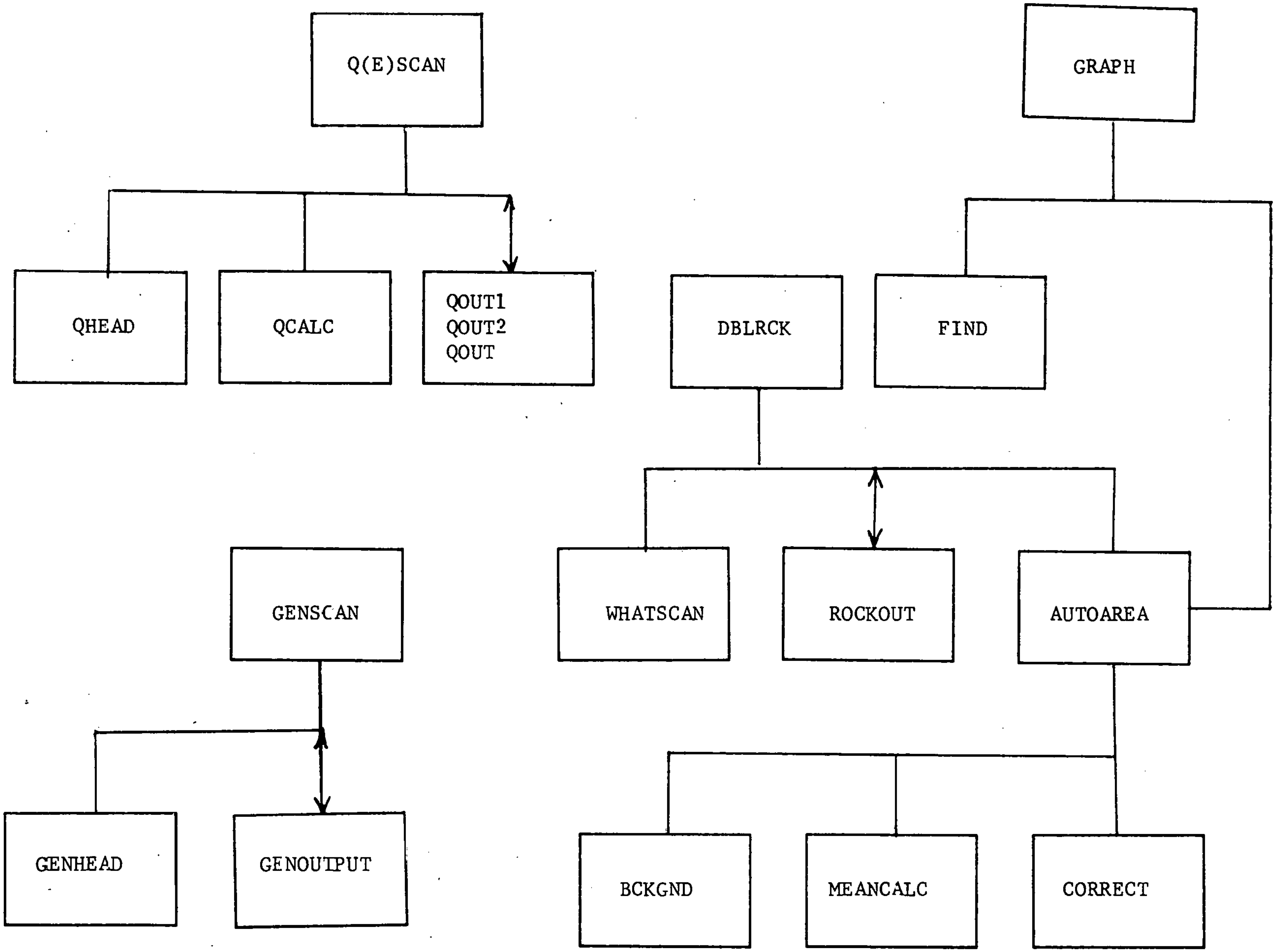


Figure 5. Task trees for driving routines and GRAPH.

a rubout character is sent to start the teletype. The task pauses 1/6 second after sending this character to allow the teletype motor to come up to speed. If this is not done, then several characters of the output line may be lost.

Routine START is called to initiate a series of scans. INPUT is called to set up the initial scan and data files. The file number on the count data set is extracted from ANS and stored in case the scans are changed after input. If a scan is being changed (BIT $\neq$ 0), then the previous scan is read unless it is the first scan on the data set. If the end of file bit is set on the previous file, the scan data set will be extended. The end of file bit is reset and the scan is rewritten to the data set.

If the scans are being updated, then the task returns control to the calling task. Otherwise if this is the last scan to be entered, then BIT is set to the scan number just entered; otherwise the next scan is entered. The values of ANS and KEYSAVE are restored before return in case this task is called again. The proper driving task is now called.

The driving task sets up the scan file starting with the file 3. It tests to see if the scan is complete. If so, the end of file bit is tested to see if that is the last scan on the file. If so, the task is terminated; otherwise the next scan file is read. The first scan that has not completed is executed.

When the scan is sent to the 910 for execution, an output task is called asynchronously, except for the Mitsubishi STEPSCAN tasks. The initial values of the scan parameters are not changed on the data set to allow scans to be repeated.

All scans initially have KEY1 set to the starting key for the data file. After the scan is complete, KEY1 becomes the starting key for the next data file. As the scans are read for a restart, KEY1 must not be updated if the scan is not complete since it contains the current output key value. When a new scan is started KEY1 is set to the starting key of the next data file which is held in location SAVE.

The Triple Axis task sends a scan as soon as it is set up. The Mitsubishis, however, must generate a character interrupt before they can receive a scan (refer to the discussion in the next section). The start commands for the Mitsubishi are scheduled upon the experiment file open. When the driving task has a scan set up, it waits for an interrupt from the 910 indicating that the Mitsubishi is ready for a scan. Normally an event variable will be set complete to indicate that the interrupt has been received.

After the scan is started by the appropriate diffractometer, the task enters a wait state. When a count is complete, or an abnormal interrupt hits, the event controlling the wait state is set complete. The interrupt oncode is then examined to determine the cause of the interrupt. The oncodes for the Triple Axis are given in Table 2, while the oncodes for the Mitsubishis are given in Subsection IV.b.

If the interrupt is a count complete, then the data are read from the 910. The event controlling the output task is set complete to allow the output to the teletype to be printed. The data are written to disk while it is printed on the teletype. The event variable which indicates that the teletype output is complete is set and the scan file is updated. The task then returns to the wait state for the next data point. After the final data point has been collected, the task goes on to the next scan.

All tasks requiring arithmetic teletype input have conversion ON BLOCKS to trap conversion errors. This keeps tasks from abending if a non-arithmetic character is entered by mistake. Also, all tasks using the count data sets have error ON BLOCKS to trap a data set seized flag since this causes the task to terminate. This ON BLOCK pauses and then the task schedules the OPEN file again. This software is required to have truely shared files as the system does not handle shared files. The scan data set is open throughout the task execution to protect it from accidently being written to while a task is executing.

DBLRCK:*

Call list parameters:

TITLE - heading to be printed at start of each scan on the teletype.
SCANIN - file containing scans to be run.

---

*Users Guide to Finger, Thumb, and Triple Axis Real Time Experiment Control Programs. T. G. Pinter and E. M. Notis, IS-4169, Ames Laboratory USDOE, Iowa State University, Ames, Iowa.

This task is the controlling task for a double rock experiment. The encoders are read to locate the positions of the arms for the Triple Axis. The number of steps for each spectrum is retrieved. The proper spectral values are set up for the spectrum being performed. If the scan is being restarted, i.e., one of the two spectra of the scan is in progress, then the spectral values are retrieved to complete the scan and the scan is sent to the 910. If a spectrum of a given scan is to be started, then the task calls task WHATSCAN to set up the proper scan parameters. On return from WHATSCAN, the data file is initiated. The spectra for the double rock are repeatedly written over each other.

ROCKOUT is called asynchronously for printing the scan on the teletype and the scan is sent to the 910. The task then enters the scan loop. Upon completion of the scan loop, AUTOAREA is called to calculate the weighted mean and the area under the spectrum. Upon return from AUTOAREA, the change in the mean from the last spectrum is calculated and compared against the convergence criterion for this scan. If the change is less than the convergence criterion, then the iteration counter is incremented; otherwise the iteration counter is reset. If the iteration counter is 2, then the scan is complete and the final means are printed on the teletype. The state of the scan is written to the scan file and the task transfers control to read a new scan or to set up a new spectrum.

DISKREAD:

This is a utility task for printing the contents of the data file on the teletype. The task requests the file to be dumped. The data set is searched for the proper file by reading the pointer for the first file to locate the next file etc. until the desired file is reached. Record 1 of the file is read. Word 1 determines which type of experiment is on the file and control is transferred to the proper print code. The initial values of the scan parameters are printed on the teletype followed by the key values and the data values.

DROCKIN:

Call list parameters:

- LOCK - word containing bits set for no arm motion.
- SCANIN - data set upon which to write the scan files.
- INKEY - key of the file entry on the scan data set.
- KEYSAVE - initial key values of the data file on the count data set.
- C - bits 0/8 contain the data file number for the first file on the count data set.
- BIT - on input its value is 1 for the scan replacement only, 0 for initial scan input, and -1 for replacement of the first scan on the file. On return it contains the key value of the last scan entered.

This task is the input task for double rock experiments. The scan type is scanned to set the proper bits in the move word for the respective angles which will move during the scan. For the Mitsubishis these are always the GA-arm and the sample table. The input values are stored on the data set and input for the next scan in requested.

DUMPSPD:

This is a utility task for dumping the semi-permanent data from the file SCANTEST. The semi-permanent data are changed on both the SCANIN and SCANTEST data sets when the SCANIN data set is updated. This allows the semi-permanent data to be dumped when an experiment is running. The file SCANIN cannot be accessed while an experiment is being executed for protection reasons.

GENHEAD:

Call list parameters:

PRESET	- counting interval for the scan.
PT1	- array containing pointers for the angles which move during the scan.
PT	- pointer to the proper heading for the scan.
ANG	- array containing the initial values of the angles.
SCNTYP	- the scan type being executed.

The scan type and heading pointer are interrogated to print the proper headings on the teletype for the Triple Axis experiment being performed.

GENOUTPUT:

Call list parameters:

PRESET	- counting interval for the scan.
OUTKEY	- the key value of the data entry on the count data set.
ANGOUT	- array containing the values of the angles to be printed.
COUNTS	- array containing the data, background, and non preset counting interval.
PT1	- pointer to the proper scan output format.
E4	- event variable for signals to STEPSCAN.
A	- number of detectors used in the scan.

This routine prints the key value, angles, and the counts on the teletype during a Triple Axis general stepscan experiment. The proper length output string is passed to the internal task PRINT for the editing of the output line. The angles are updated and printed as each data point complete is received from the 910.

GETSCAN:

Call list parameters:

SCANIN - data set to write the scans.
ANS - bits 0/8 contain the data file number for the first scan, bit 17 is set for writing descriptions of each line of scan count data set.
INKEY - key of the file entry on the scan data set.
KEYSAVE - initial key value of the data file on the count data set.
BIT - on input its value is 1 for scan replacement only, 0 for initial scan input, and -1 for replacement of first scan on the file. On return it contains the key value of the last scan file entered.

The appropriate file of the semi-permanent data is read to retrieve the plane of the scan and the Triple Axis $\vec{q}$ indices for the 2-dimensional vector. The appropriate bits in PRIN_EN for Triple Axis are set and $\vec{q}$ is reduced from the 3-dimensional input vector to the 2-dimensional scan vector and the scan is written to disk.

INPUT:

Call list parameters:

EXPTYPE - experiment identifier.
TITLE - heading for the teletype output.
SCANIN - data set containing the scan files.
LOCK - word containing the bits set to hold arms fixed throughout the scan.
BIT - set if count data set is not to be accessed.
E1 - event variable to allow file 0 of scan data set to be written.

This task controls the input for all the experiments. The Triple Axis code sets up LOCK. The bits 9/17 of LOCK are set if the respective θ_M - θ_{A2} arm is not to move during the scan. The starting data file on the count data set is then entered on the teletype. The data files are searched to locate the starting key for the data file for this scan, KEYSAVE. KEYSAVE is then passed to the input task for writing on each scan file.

The three letter code for the experiment type then determines which input task must be called. After return from the input task, any given file can be corrected by entering the desired file number. If the last scan on the file is to be replaced, then BIT is set so that the end of file bit will be reset, and additional scans can be read; otherwise BIT is set so that only the specified scan will be replaced. INKEY is set to the key of the scan file to be rewritten. Control is then passed to the proper input task. Finally the new TITLE is written onto the count data set.

INPUTGEN:

Call list parameters:

LOCK - word containing bits set for no arm motion.
SCANIN - data set upon which to write the scan files.
INKEY - key of the scan file.
KEYSAVE - initial key values of the data file on the count data set.
STPSZE - on input bits 0/8 contain the data file number for the first scan on the count data set.
BIT - on input its value is 1 for the scan replacement only, 0 for initial scan input, and -1 for replacement of the first scan on the file. On return it contains the key value of the last scan entered.

This is the input task for a general scan experiment. The scan type is scanned to determine which bits in the move word should be set in SCNTYP. Bits 7 and/or 8 of SCNTYP are set if detector 2 and/or detector 1 are to be used. SETUP is an internal task used to set the proper angle bit in SCNTYP.

<u>SETUP</u>:

Call list parameters:

- I - the pointer of the angle to be moved in the scan.
- K - counter for the number of angles moving in the scan.

This internal task in the Triple Axis code computes the initial scan angle for angles moving during the scan, stores the angle index in the array for angles which move, and sets the proper bit in SCNTYP. The angle increment is assigned for the first two angles of the scan which move.

<u>INPUTSP or INPUTO</u>:

Call list parameters:

- SCANIN - file used by calling task.
- TEST - line number for semi-permanent data entry.
- ANS - set for writing descriptions of each line of semi-permanent data.
- WRITE - set to update file SCANTEST.

This task allows input of the semi-permanent data. All or part of these data can be entered during any task call.

The current semi-permanent data are read upon entry into the task and the current calibration vector indices are established.

Control is transferred to the requested line number for data input. After all of the semi-permanent data have been entered, the plane of the scan is determined by testing the possible values of the denominators of a and b (Eqs. 3). The $\vec{e}$ unit vectors are assigned and the plane identification is assigned. The indices of the 3-d $\vec{q}$ are packed into SCAN in the Triple Axis codes. The magnitudes of the calibration vectors are calculated using Eqs. 1. ETA is calculated using Eqs. 4. The semi-permanent data are written on the proper scan file. If the semi-permanent data are being changed during a run, they are also updated on the data set SCAN TEST.

INTTEST:

Call list parameters:

J	- interrupt vector oncode.
TRIAX	- experiment file.
ANGIN	- array containing current angles.

This task handles abnormal interrupts for the Triple Axis diffractometer only. See Table 2 for the ONCODES. The 6 low order bits containing the angle pointer are extracted from the oncode. The task tests to see if bit 11 is set. If so, then the encoders are read and the angle causing the overshoot, along with its current and desired values are printed on the teletype. If bit 15 is set then the bumper limit error is printed. If bit 16 is set then the panic error is printed. If bits 9 or 10 are set, then the high or low limit switch error is printed along with the arm that hit the switch. These are external limit switches physically attached to the diffractometer. They are not to be confused with the internal limit switches in the semi-permanent data. The execution of the driving task is halted when any of the later four interrupts are detected; otherwise control is returned to the calling task.

<u>QESCAN</u>:

Call list parameters:

TITLE	- heading to be printed at start of each scan on the teletype.
SCANIN	- file containing scans to be run.
INKEY	- key of current scan file being processed.

This task is the driving task for a q(E) scan. Since $\vec{q}$ and ν determine the angle settings, a given scan is treated as N2 separate scans, each scan having N1 points. The N1 data points are collected at one angle setting since the counting times are long. This allows the data to be checked statistically for abnormal fluctuations in the count. The angles are stored on disk as they are calculated. The value of NOS depends on the particular driving task. Triple Axis code selects a different output task for N1=1 and N1 $\neq$ 1. NOS points can be calculated at any one time. Thus, a given experiment may consist of several partial scans.

Task QHEAD is called to set up teletype headings and to initialize the count data file. On return, the scan being performed is read by the 15. The number of steps left in the scan is extracted from NSL. This value is reduced to the modulo variable NSC for the partial scan. NOS is the number of angles to be calculated in Q(E)CALC. If NSC is zero, then the next set of angles must be calculated; otherwise some remain unprocessed from the previous run.

The proper output task is called asynchronously. These tasks handle the teletype I/O as well as testing the statistics of multiple counts at one point.

The scan loop processes NOS steps. The angles are read and tested to see if any of them have exceeded their internal limits. If so, then the point is skipped. The encoders are read to compute the current positions of the arms. If detector 1 is being used, angles $2\theta_{S2}$-$2\theta_{A2}$ are always fixed. Similarly angles $2\theta_{S1}$-$2\theta_{A1}$ are fixed for detector 2. Next the angles are tested for backlash. The angles which do not move during the scan are checked for drift before each new scan is sent to the 910. If they are more than 1 hundreth of a degree from their calculated position, then they are backlashed and returned to their calculated position. If the angle is being scanned, then backlash is only done when the angle moves in the negative direction.

This is the start of the count loop for N1 counts. When a count complete interrupt hits, the data are written on the partial count file. This allows the partial counts to be retrieved if the task is stopped during the partial scan. This is the end of the count loop.

If the point is statistically acceptable, then the sum is written in the appropriate record of the data file.

Upon completion of the scan loop, the output task is exited. If there are points remaining in the full scan, control is transferred to Q(E)CALC. If the end of the data set bit is reset, then control is transferred to start of an experiment; otherwise, the program is terminated.

Q(E)CALC:

Call list parameters:

NS	- number of angles to be calculated.
INKEY	- key of the current scan file.
SCANIN	- data set containing the scan files.
KS	- starting key for storing calculated angles on data set ANGSTR: 0 for run; test value depends on the experiment.
NSL	- number of steps completed in the scan.

This task calculates the angles of the scan from the wavevectors in reciprocal space.

The semi-permanent data and the scan file are read into their respective structures. Since the components of $\vec{q}$ and ν are not updated, the current values must be calculated. Cosη and sinη are also calculated. The Triple Axis $\theta/2\theta$ pair which will not vary during the scans is calculated using Eq. 9. This starts the angle calculation loop. Compute $\vec{q}$ for this scan point using Eq. 6. Compute $2\theta_{S1}$ using Eqs. 8. Compute Ψ using Eqs. 7 and 8. Compute the varying $\theta/2\theta$ pair. Write the angles to disk and repeat the angle calculation for the next point.

If both Triple Axis detectors are being used, compute $2\theta_{S2}$ using Eqs. 8 and the $\theta_{A2}/2\theta_{A2}$ pair using Eq. 9. The inelastic scattering phonon energy for detector 2 is computed. The angles and phonon energy are written to disk for use by task Q(E)SCAN.

The following internal tasks are used in Q(E)CALC.

CALANG:

Call list parameters:

- I - used for calculation of the momentum vector. If set, the k'=sqrt(1.995998E); else k'=sqrt(K).
- E - energy used for momentum vector calculation.
- K - square of momentum vector, used if I is reset.

This task calculates the scattering angle using the Bragg Law (Eq. 9).

ANGST:

This task tests the calculated angles against their internal limits and adds the zero reference angles. The task tests to see if scattering angle 2θ is being tested. If so, then the θ angle is calculated using the relationship, $\theta=\theta_z+"2\theta"/2$ and stored in the angle array. Then the zero reference is added to the calculated angle.

Now the angles are checked against their internal limits. The angle is checked against its low limit. If it is less than the low limit, then 360 degrees is added to the angle value. This is done to allow angles to pass through 0 degrees. If the angle value now lies between the upper and lower limits, it is accepted and stored modulo 36000 in the angle array. If the angle is not acceptable, then -1 is stored as the value of $2\theta_{A2}$ to mark the scan point invalid.

QHEAD:

Call list parameters:

HEADR	- heading of scan output on teletype.
TOM	- bit set for count against monitor, reset for count against time on return.
Q	- vector containing $\vec{q}$ and ν components and increments along with the fixed energy on return.
INKEY	- key value for current scan file.
DATAS	- array of partial counts on return when restarting a run.
SAVE	- the key for the first record on last data file on call and the key for the first record on data file on return.
DATA	- prescale multiplier value.
N1	- number of counts at one angle setting on return.
N2	- number of steps in the scan on return.

This task prints the heading information for the scan on the teletype. It sets up the proper values for the q components in 3-space as well as initializing the data file on the count data set. Various headings are printed. $\vec{q}$ in 3-space is calculated from the 2-space representation using the indices in INDCE or TAG.

If the experiment has been restarted, then the partial count array is read into DATAS to retrieve the data counted to this point. If the scan is initially being started, then the data file on the count data set is initialled. Value SAVE is set to the key of the first data record.

QOUT1 or QOUT:

Call list parameters:

COUNTS	- array containing the data counts.
KEY1	- file key for writing to disk.
Q	- array of $\vec{q}$ components and energy.
ANGLES	- array containing output angles.
E1	- event variable for signals from Q(E)SCAN.

E2 - event variable for signalling Q(E)SCAN.
PRIN_EN - bit variable containing detector and fixed energy bits.
TOM - time or monitor bit.
NH - number of counts completed at one setting.
NC - number of total counts at one setting.
NSL - number of steps completed in the scan.
NS - number of total steps in the scan.
DATAS - array containing the partial count data.
SUM1 - sum of the data from detector 1.
SUM2 - sum of the data from detector 2.
BR - branch parameter: set for final print of heading information at scan start or restart and reset for print of count data only.

QOUT1 controls the teletype output and statistical check for multiple counts at a given point for Triple Axis diffractometer. QOUT controls the teletype output for all count loops as well as the statistical check for multiple count loops for the Mitsubishi diffractometers.

The initial angles and final headings are calculated. The task then waits for the start of the scan loop in Q(E)SCAN. NCF is the save location for NC since this value will change if a count is repeated. The $\vec{q}$, energy, and the varying angles are printed as a heading for each set of point counts at a given step.

The count loop begins here. The data, background, non-preset time interval, and point number are printed on the teletype. This is the end of the count loop.

After completion of the count loop, the data are checked statistically. This is done in case a spurious data value is counted for some reason. The partial counts are summed and the standard deviation of the sum is calculated. If the sum is greater than 99, then the test value is 5 standard deviations; otherwise it is 3 standard deviations. Each data point is then checked against the average count. The largest difference between the average count and the partial data is marked by setting NH to its array pointer. If this difference is greater than the test value, then a corrected sum is calculated; otherwise NH is set to zero. The key value on the data file, the sum, and the corrected sum are printed on the teletype. Q(E)SCAN is then signalled that the statistical check is finished. If a point is to be recounted, control is returned to the partial count loop; otherwise control is returned to the scan loop. The sums and corrected sums are then printed on the teletype. Q(E)SCAN is then signalled that its processing can continue. This is the end of the scan loop.

QOUT2:

Call list parameters:

COUNTS	- array containing the data counts.
OUTKEY	- file key for writing to disk.
Q	- array of $\vec{q}$ components and energy.
ANGLES	- array containing output angles.
E1	- event variable for signals from Q(E)SCAN.
E2	- event variable for signalling Q(E)SCAN.
PRIN_EN	- bit variable containing detector and fixed energy bits.
TOM	- time or monitor bit.
NSL	- number of steps completed in the scan.
NS	- number of total steps in the scan.
NH	- number of counts completed at one setting.
SUM1	- sum of the data from detector 1.
SUM2	- sum of the data from detector 2.
BR	- branch parameter: set for final print of heading information at scan start or restart and reset for print of count data only.

This task outputs count data to the teletype when there is only one count per angle setting. The task prints the key value, $\vec{q}$ components, ν, and selected angles on one line.

<u>RESET</u>:

Call list parameters:

SCANIN - data set upon which to write the scan files.
SCAN - type of experiment being performed.
LOCK - word containing bits set for no arm motion.

This task is called by UPDATE to update the current scan files. Four types of operations are possible: repeat of a completed scan; addition of scans to the end of the present files; replacement of any scans currently on the file except for those already complete; and deletion of any scans yet to be run.

If a scan is repeated, the number of steps completed is set to 0 and the data file key is reset to the start of the file. The task is then exited.

In order to add to the end of the scan data set, the last scan file on the data set is located. The end of file bit is reset, and the proper input task is called with BIT set 0.

In order to delete a scan file, the file is read into core. The number of steps completed is set to the number of steps in the scan if general scan or q(e) scan. The proper bit in EOF is set if the scan is a double rock. The key of the data file is set to the starting key of the next data file in case the scan was in progress when the deletion occurred. The data file numbers must be reduced by one on each succeeding scan file on the data set after deletion unless the deleted scan was the last scan on the file or the scan was in progress when deleted.

To replace a scan, the proper scan file is read into core. If it is the last file on the data set, then the data file number is reduced by one, the seven low order bits are extracted from EOF, and BIT is set to zero to add a series of scans. Otherwise BIT is set to 1 to replace only the scan in question. The proper input task is then called.

Upon completion of all desired input, control is returned to update for starting the proper driving task.

RESTART:

This task reads the data file to determine which driving task was being used upon experiment termination. Control is then transferred to this task and the experiment continues from the point at which it was interrupted.

ROCKOUT:

Call list parameters:

DATA	- array containing the data counts.
DTOUT	- the data point to be printed on the teletype.
TM	- the nonpreset counting interval.
N	- key of the output data file.
E2	- event variable for signalling DBLRCK.
E1	- event variable for signals from DBLRCK.
M	- number of steps left in the scan.
A1	- value of the primary scan angle.
A2	- value of the secondary scan angle when scans are coupled.

ANG1 - pointer to the first scan angle.
ANG2 - pointer to the second scan angle.
TOM - bit set for counting against time and reset for counting against monitor.

This task prints the initial angles of the scan upon entry. The key of the data file, the varying angles, and the data count are printed on the teletype. When the scan is finished, the task is exited.

START:

This task is the initial driving task for the diffractometers. It initializes the data file for the type of experiment currently being executed as well as the directory identification characters.

The task calls task INPUT for getting proper experimental input data. INPUT returns TITLE which is written in records 1/11 of the data file. This allows the current TITLE to be preserved for future runs. The experiment word, the angle fixing word, and TITLE are then saved in record 0 of the scan file for Triple Axis. TITLE is written in files 0 and 1 of the scan data set for the Mitsubishis. The rest of this section will describe the individual tasks in the task tree. The call list parameters will be given along with a brief description of the details of the task. The proper driving task is then called.

STEPSCAN:

Call list parameters:

TITLE - heading to be printed at start of each scan on teletype.
SCANIN - file containing scans to be run.

This task finds the first scan that is not finished. It sets up the experiment data set. The array of angles to be moved is scanned. If the angle is 0, then the current value of that angle is used in the scan. If the angle is 360, then it is set to 0 for the scan.

If the task is a restart (NSL ≠ 0), then the current value of the angles are calculated from the initial values stored in the scan file; otherwise the data file is initialized. When task GENHEAD has completed, then GENOUTPUT is called asynchronously to print the data to the teletype. The loop for receiving data points is then entered. After each count, the angles to be printed are incremented, and the number of steps completed is incremented on the scan file for restart purposes. When the scan is completed, control is transferred to read the next scan, unless this was the last scan to be performed. In the Mitsubishi codes, a scan of greater than 99 points must be broken into several smaller scans if the scan is automatic since a 99 point scan is the longest automatic scan which can be executed.

TESTO(E):

This task computes angles for a trial q(e) scan and prints them on the teletype. The semi-permanent data are copied into the lower files to allow for update when testing without destroying the current copy. If the semi-permanent data are to be changed for the test, the INPUTSP is called with I set to zero so that only the copied semi-permanent data will be updated. Routine GETSCAN is then called to get the scans to be tested. The angles are computed in groups and are printed.

UPDATE:

This task calls RESET to allow the scan file to be updated. Control is then transferred to the driving task which was executing when the experiment was interrupted.

WHATSCAN:

Call list parameters:

D910 - structure containing the 910 experiment file.
ANG1 - initial value of scan angle one.
ANG2 - initial value of scan angle two.
INKEY - key of the file entry on the scan data set.
SCANIN - data set containing the scan to be run.
NO1 - pointer to the first angle of the scan.
NO2 - pointer to the second angle of the scan.
SAVE - temporary storage for the starting key.

This task sets up the proper scan for a double rock, initializes the data file, and performs a quick scan at every other scan point if PRESET is zero. The proper scan is read into core. If OUTKEY is 0, then the scan has not been started. The data file pointers are written on the data file.

The angle values are next compared with their present positions. The angle pointers are retrieved from START. The proper scan values are set up depending whether this is scan 1 or scan 2 of the experiment. If PRESET is nonzero, then contról is returned to the calling task.

If PRESET is zero, then the increment and number of steps are divided by 2. A scan is sent to the 910 and the task enters its wait loop. The first and last scan points are stored. The maximum data count is also stored. The value of the PRESET is then set so that the maximum counts will be 2000. The difference between the first and last scan points is calculated. If this difference is greater than 100, then the stepsize of the scan is multiplied by MULT. Control is transferred to set up experiment file.

B. The Plotter Codes

This section describes the codes to generate a plot on the x-y plotter. The controlling task is called GRAPH. It contains options for calibrating the plotter and checking this calibration, plotting a set of data stored on disk, and calculating areas either automatically or with the spectral features input by the user. The codes which perform the plotting will be discussed here while the area calculating codes will be discussed in the next subsection.

<u>FIND</u>:

Call list parameters:

- SCNUMBR - the number of the data file on the count data set.
- START - the starting key of the desired data file.
- NS - the number of data points in the scan.
- INCR - the increment for locating the data counts. This increment is always 1 unless 2 detectors are used with the Triple Axis diffractometer.
- ANGLE - the initial scan parameters on return.
- STEP - the angle or q stepsize of the scan on return.

File 0 of the count data set is read to determine in which directory the data set resides. The proper offsets are then set up for the data file to be read. The count data set is then searched for the proper data file to be processed and the scan parameters are read from the file. The start and stop keys are computed from the data set keys. If the data set is in directory TRIAX, then if 2 detectors were used, a request is made for which detectors data are desired. The proper values are then set up as above.

GRAPH:

This task is the driving task for the plotter. The option desired is tested. If the option involves the plotter, the proper key and test value are set up for a write to plotter. A test value of -1 enables the plotter interrupt, whereas a test value of 0 disables the plotter interrupt. TEST is 0 for a plot so that any calibration or checking operation will be halted. The plot experiment file is written from the appropriate key value. If the option was calibration or check, then control is immediately returned to get another option. The plot option requests the data file to be plotted. FIND is called to locate the starting and stopping key values for the data and the scan parameters. The scale factor and range are calculated such that the maximum y-value plotted is 1023. The data points are read from the data file in groups of 12 to minimize core storage. They are scaled and sent to the plotter to be plotted. After each group of 12 points is plotted, the total number of points sent is compared to the number of points on the file. The 12 point plot loop is repeated until the entire file is plotted. After each

12 point send, the task waits for an interrupt from the plotter that the plot is completed before continuing.

C. The Area Calculation Routines

These tasks are used both by the double rock tasks and the plotting tasks to calculate the area under a given spectrum along with its weighted mean. The mathematics used in these calculations is described in Section II.B.

<u>AUTOAREA</u>:

Call list parameters:

M	-	$|M|$ is the number of points in the scan. $M > 0$ for angle scans and $M < 0$ for $\vec{q}$ scans.
NO	-	starting key of the data file to be processed.
MEAN	-	the weighted mean on return.
KT	-	parameter is set for automatic background search; otherwise reset for user input of background features.
DPHI	-	the x-increment of the spectrum.
PHIO	-	the initial x value of the spectrum.
INCR	-	key increment for reading the data file.

The task reads the data file into core. If KT is reset, then the spectral features: the start and end of the backgrounds and the start and end of the peak are entered on the teletype. If KT is set, then BCKGND is called to compute these spectral features. The position of the maximum count of the spectrum is computed and its position is marked. If the background exists on both sides of the spectrum, then a straight line is fit to the background and subtracted from the spectrum using CORRECT. Finally the weighted mean of the spectrum and the areas are calculated in MEANCALC.

BCKGND:

Call list parameters:

Y	- the array containing the data counts of the spectrum.
ELB	- the end of the left background.
CL	- the left end of the peak.
CR	- the right end of the peak.
ERB	- the end of the right background.
NO	- the number of points in the spectrum.
IS	- the pointer to the location of the maximum data count.
PHIO	- the initial x-value of the abscissa.
DPHI	- the x-increment.
KEY	- the starting key of the data on the data file.

This task computes the background features of the spectrum. A value 90% of the maximum value is calculated. The spectrum is search on the left and the right for the salient features. The position of the peak is printed on the teletype before the task is exited.

SEARCH:

Call list parameters:

PT1	- start of background.
PT2	- end of background pointer.
PT3	- start of peak pointer.
K	- key increment for y array.
PRINT	- alphanumerics for printed background keys.

This internal task is the search task for the background. The average standard deviation of the first 3 points, ALPHAS, is calculated. The difference of the first 3 points and the second 3 points is compared to ALPHAS. If this difference is greater, then a counter is incremented; otherwise the counter is reset. When the counter reaches 3, the background end is marked. When the counts reach the 90% value of the maximum count, the peak position start is marked. If a background was determined, then the average count is printed along with its key position.

CORRECT:

Call list parameters:

Y	- array of data points.
SLB	- start of left background pointer.
ELB	- end of left background pointer.
ERB	- start of right background pointer.
SRB	- start of right background pointer.

This task fits a straight line to the background points on each side of the peak. The slope and intercept are then used to calculate a line to be subtracted from the spectrum. This removes the background from the area and mean calculations.

MEANCALC:

Call list parameters:

Y	- array of data points.
ELB	- end of left background pointer.
CL	- start of peak pointer.
CR	- end of peak pointer.
ERB	- end of right background pointer.
PHIO	- initial x-value.
DPHI	- x-increment.
MEAN	- the position of the maximum data count on entry and the weighted mean of the spectrum on return.
J	- adjustment from array pointer to key value.

This task calculates the mean and the area under a given spectrum. The area is calculated by a direct summation of the data points and also by Simpson's rule. If the peak positions are greater than 7 keys apart, then the task searches for the edges of the plateau. All points within 3 standard deviations of the average count of the original peak positions are considered in the plateau. The mean is just the center of the plateau. The weighted mean is converted to the scan parameters. If the mean is too far from the maximum point of the scan, the maximum point of the scan is taken as the mean. The mean and areas are printed on the teletype.

D. The Tape Punching Codes

In this subsection the tape punching codes will be described. Some of the tasks used are also used in the driving codes. They will not be discussed here.

DUMP:

Call list parameters:

KEY - the number on scans on the tape.

This task prints the scans that are written onto the paper tape for a manual run of the Mitsubishi.

CHARGEN:

Call list parameters:

A - the GA-arm angle setting.
B - the VA-arm angle setting (FINGER only).

C	- the sample table angle setting.
D	- the angle increment for GA-arm scan.
E	- the angle increment for VA-arm scan (FINGER only).
H	- number of steps for GA-arm scan.
I	- number of steps for VA-arm scan (FINGER only).
KEY	- key of tape file to write the character string.

This task generates the character strings to be punched on the tape and writes them out to disk to be punched later.

<u>TAPEQ</u>:

Call list parameters:

SCANIN	- file containing the scans to be punched onto paper tape.
HEADR	- heading for the teletype output.

This task calls Q(E)CALC to generate scan angles for a q(e)-scan. The angles are checked for backlash and a backlash scan is generated if needed by CHARGEN. The scan point is then generated by CHARGEN. A separate scan must be generated for each point of the scan. The values of the angles and the $\vec{q}$-components are then printed on the teletype. Whenever the data set containing the scans to be punched is full, TAPEPNCH is called to punch these scans and DUMP is called if the scans are to be printed on the teletype.

<u>TAPESCAN</u>:

Call list parameters:

SCANIN	- file containing the scans to be punched onto paper tape.
HEADR	- heading for the teletype output.

This task generates the scans to be executed for a general step scan. The scan angles are printed on the teletype as well as the increments of the scan. A backlash scan is set up before the scan is executed. If the number of steps to be executed is greater than 99, then the scan is broken up into shorter scans as 99 steps is the maximum for an automatic scan. If the scan is a sample table scan, then one scan is required for each point of the scan as the sample table scans are not automatic. Whenever the data set containing the scans is full, then they are punched by TAPEPNCH and written to teletype by DUMP if desired.

TAPESTART:

This task is the driving task for the tape punching routines. INPUT is called to get the scans to be punched from the teletype. The proper task is then called to punch the scans.

TAPEPNCH:

Call list parameters:

- BRANCH - branch for experiment type--set for FINGER, reset for THUMB.
- PTAPE - file containing the scans to be punched on the paper tape.
- TEST - if set, then an end of tape character is punched after the last scan.
- KEY - key of the last record on the data set to be punched.

This task calls SETUP with the proper parameters for the FINGER or the THUMB.

SETUP:

Call list parameters:

STRING - input string for reading the scans from the tape data set.
I - number of words in the string.

This task reads the tape data set and converts the scan to ASKII code for punching on the paper tape. The characters are stored in the array BUFFER which is punched on the paper tape one character at a time. After the first character is punched, an interrupt is received. If more scans are on the data set, then the next scan is converted; otherwise the task tests whether this is the last scan on the tape. If so, then the end of tape character is punched onto the paper tape.

IV. PROGRAMMING ALGORITHMS FOR THE 910

In this section the algorithms for the 910 diffractometer routines are described. There is a separate routine for driving each experiment: the Plotter, Triple Axis, Finger, and Thumb. The algorithm for each driving routine will be described in PL/I like structures using English phrases. The constants and internal entry points will be defined. All values and labels occurring in the 910 code explicitly are in capital letters in the algorithms; otherwise, they are in small letters. Octal numbers are represented by the suffix (8) appended to any number, and decimal numbers are represented by the suffix (10).

A. Plotter

The Plotter sends a point to be plotted through address 167(8) of its interface with the 910. When this point is plotted, an interrupt is sent to the 910 through this interface. This interrupt is trapped with subsequent action being determined by the software interrupt handler.

The (x,y) coordinates of the point to be sent are packed into one 24 bit word. The leftmost 12 bits contain the y-value, and the rightmost 12 bits contain the x-value, both in integer form. The maximum value that can be plotted is 1023(10) and is subtracted from 1023(10) or 1777(8) before it is sent to the Plotter. Thus, the origin would be 17771777(8). A point at (5,10) would be represented as 17721765(8) when sent to the plotter.

The calibration mode outputs 12 points at the origin. This is followed by 12 points at (x(max),y(max)). The slow clock is scheduled between each send for timing purposes.

The check mode outputs a point at the origin, (x(max),y(max)), and the origin again. The slow clock is scheduled between each send also.

The plot mode plots the points in groups of twelve at a time. The next point is sent as soon as the interrupt hits from the previous point.

The interrupt handler tests location INT before processing the interrupt. If INT=0, then the interrupt is ignored. This effectively stops any operation in progress. The value of INT is -1 when the interrupt is to be processed.

The following entry points and buffer areas exist for the 910 plot code.

PLTINT: Entry point for Plotter interrupt handler.

ONOFF: Entry point for code execution upon an open or close of the experiment file PLOT.

CLBRTE: Entry point for the calibration of the x-y Plotter upon WRITE (PLOT) KEYFROM(1). A -1 is stored in location INT when the write is executed to enable the interrupt handler.

INIT: Entry point to initialize the plot upon WRITE (PLOT) KEYFROM(2). SCALE is the range of the abscissa and NS is the number of points in the spectrum. These number are passed from the 15 when the write is executed.

PLOTXY: Entry point to plot twelve points at a time upon WRITE (PLOT) KEYFROM(3). The points are stored in array Y when the write is executed in the 15.

PURGE: Entry point to stop calibration or check of Plotter upon WRITE (PLOT) KEYFROM(4). INT is set to 0 when the write is executed.

CHECK: Entry point to check calibration upon WRITE (PLOT) KEYFROM(5). INT is set to -1 when the write is executed to enable the interrupt handler.

The algorithm for the Plotter is given below:

```
CHECK:
  branch to proper entry (initially NOP)
  mark task check--set TEST -1
  set up task reentry point--location ADDR
  initialize check counter--set COUNT 2
CHK:
  store branch CHK2 in location BRNCH2
  store origin for output to Plotter
ℓ1:
  output point to Plotter
  decrement check counter COUNT
  store return branch CAL0 in location BRNCH1
    then return to system
    else do
      dissable interrupt handler--set INT 0
      reset BRNCH2 to NOP
      return to system
    end
CHK2:
  store branch to CHK1 in location BRNCH2
  store maximum for output to Plotter
  branch to ℓ1
CLBRTE:
  branch to proper entry (initially NOP)
  mark task calibration--set TEST -1
  set calibration counter--set TEMP+1 -1
  set up task re-entry point--location ADDR
  store return branch CAL0 in location BRNCH1
  store origin for output to Plotter
ℓ3:
  initialize point counter--set COUNT 11
CAL0:  re-entry point
  decrement point counter
  send point to Plotter
  if more points to be sent--COUNT >=0
    then return to system
    else
      if calibration is finished--TEMP+1=0
```

```
        then do
          store NOP in location BRNCH1
          disable interrupts--set INT 0
          return to system
        end
        else do
          mark maximum point for send to Plotter
          set TEMP+1 0
          branch to ℓ3
        end

INIT:
  convert range to floating point storage
  convert maximum plotting voltage to floating point
  divide voltage by range to get the scale factor
  initialize the partial point counter--set COUNT -12
  set up task re-entry point--ADDR
  mark task plot--set TEST 0
  enable interrupt handler--set INT -1
  send interrupt to 15--no oncode necessary
  return to system

PLOTXY:
  decrement number of points in plot
  if all points are plotted--N3 >=0
    then do
ℓ2:
      reset partial point counter--set COUNT -12
      send interrupt to 15 for more points
      return to system
    end
    else do
      if 12 points have been sent--COUNT=0
        then branch to ℓ2
        else do
          load index value with COUNT
          increment the partial point counter
          load Y-value thru index register
          subtract Y-value from 1777(8)
          shift Y-value to left-most 12 bits
          merge the X-value to complete word
          subtract the X-increment from abscissa
          convert to fixed point for next send
          send the point to be plotted
          return to system
        end
```

```
ONOFF:
  NOP BRNCH1 and BRNCH2
  return to system

PLTINT:
  if interrupt disabled--INT=0
    then return to system
    else
      if task plotting--TEST=0
        then schedule task PLOTXY
        else schedule slow clock
  return to system
```

B. Finger and Thumb Codes

The 910 starts up the Mitsubishi, sends a string of characters for the scan to be performed, and reads the data through the Mitsubishi/910 interface.

The I/O registers for the Finger and Thumb are denoted by the octal address A(xy)B(z) where A(xy) is 12 for the Thumb and 13 for the Finger. The B(z) address has the same value for both the Thumb and Finger. These values are given in Table 1. Further discussion of the algorithm will use the Thumb as an example. The basic difference between the two codes is the additional table driven by the Finger diffractometer.

On an OPEN FILE (THUMB) in the 15, entry point SETUP is scheduled. This starts the initial start up of the Mitsubishi. The code outputs a clear Mitsubishi to the interface which resets the Mitsubishi hardware.* Eighteen clock ticks are counted to

*W. D. Thomas, Private Communication.

Output Registers:

B(z)	Function
0	Output monitor counting interval
2	Output time counting interval
3	Output counting interval to hold register
4	Output character to Mitsubishi
5	Control address
	1 - set count on monitor
	2 - set count on time
	10 - clear Mitsubishi
	40 - start Mitsubishi
	400 - send master reset to interface
	Input Registers:
0	Load data register
1	Load monitor register
2	Load time register
5	Test for cause of interrupt
	1 - EOM interrupt
	2 - EOT interrupt

Table 1. I/O registers for communication with the Mitsubishi.

allow the buffer time to be read since the interface hardware is much slower than the 910 hardware. The task then outputs a master reset to the interface which clears the interface and puts the sequencer in ready for next block. Again eighteen clock ticks are counted. Finally, the task outputs a start Mitsubishi command. This last command causes a reader read which puts the interface in the block loading state. It also generates a read set which loads a zero character into the Mitsubishi. The interface then generates a ready next character interrupt.

After the first character interrupt is received in the 910, an interrupt is sent to the 15 requesting a scan. This scan is converted into a character string, for example,

N01A12050C01220D05H10M20 (THUMB),

for transmission to the Mitsubishi.

The first seven bit character is loaded into the character buffer and then transferred to the Mitsubishi. With each character output, another ready next character interrupt is generated. After the last scan character is sent, a carriage return is sent to signal the end of the block. This puts the Mitsubishi and the interface into positioning mode. The character interrupt generated by the carriage return is ignored by disabling the character interrupt handler before this character is sent.

If a character sent to the Mitsubishi cannot be properly interpreted, a non-defined character interrupt is generated in place of the next character interrupt. This interrupt is sent on the shared data interrupt line. It is determined by testing to see if the character interrupt is enabled. If a non-defined character interrupt occurs, a start is sent to the Mitsubishi, and an attempt is made to send the string again. After 6 unsuccessful attempts to send the character string, an interrupt is sent to the 15 to abort the current scan.

After the arms are positioned, a scalar start signal is sent to the interface by the Mitsubishi and counting is started. Subsequent scalar start signals will allow counting to begin only if the counting registers have been loaded before positioning is complete. This is done automatically from the H or hold register. This register is loaded with the proper counting interval before the character string is loaded. Since counting is done up to zero, the negative value of the counting interval is loaded into the H register.

When a count is complete, an end of measurement (EOM) is sent to the Mitsubishi <u>via</u> the data interrupt line. This interrupt is determined by testing address A(xy)5. When the data are read, the H register is loaded into the proper count register and the EOM flip-flop is reset. This allows positioning to start for the next point in the scan. Only GA and VA scans are performed automatically by the Mitsubishi. Sample table scans must be processed as n separate scans by the 910 software. If the scan is complete, a reader read and read start are also generated. This causes a ready next character interrupt to be generated. When the new scan is received, the conversion and loading proceed as above.

At the beginning of each scan and for all negative angle increments, the arms are first driven to .3 degrees less than the desired angle. This is done to remove backlash from the gears. One time count is loaded into the time register and the count is started. The immediate EOM is ignored except to generate a next character interrupt for the transmission of the actual scan to be performed.

After each EOM, the data are converted to 15 floating point representation. The 15 is signaled that data exist to be read. The data are read by the 15. If the scan is complete, an interrupt is sent to the 15 requesting a new scan. Otherwise, the software is exited until another data interrupt is received.

Finally, the interface has a panic or dump button which generates an end of tape (EOT) interrupt. This interrupt is also sent *via* the data interrupt line. Upon receipt of the interrupt, the Mitsubishi is cleared which stops any action in progress. A signal is sent to the 15 to terminate the task currently executing.

When the 15 terminates a task, either normally or abnormally, a semi colon is sent to the Mitsubishi to signal the end of the "paper tape". This causes an end to be generated by the Mitsubishi which disables the hardware until a new task is started in the 15.

The following entry points and buffer areas exist for the 910 Mitsubishi code.

CHINT: character interrupt handler.

DTINT: data interrupt handler.

SETUP: entry point for code execution upon open file THUMB.

SHTDWN: entry point for code execution upon close file THUMB.

THUMB: entry point for scan set up upon WRITE (THUMB) KEYFROM (1). This entry point is also used for internal task scheduling after various interrupts have been processed.

SNDATA: entry point for data transmission to the 15 upon READ (THUMB) KEY(2). DATA contains the data in 15-floating representation.

```
THUMB:
  branch to proper entry point--initially INIT
INIT:
  send clear to Mitsubishi
  store branch SND400 in location ADRES

LAB1:
  initialize clock tick counter--set TEMP 18
  schedule medium clock
  return to system

SND400:
  store branch SND10 in location ADRES
  send master reset to clear interface
  branch to LAB1

SND10:
  send start to generate character interrupt
  store branch INT in location ADRES
  store oncode for interrupt--set TEMP 1
  return to system
```

```
INT:
  store branch SCAN in location ADRES
  send interrupt to 15
  return to system

START:
  enable character interrupt handler
  set number of steps--set CHAR+6 to 22030061(8)
  set angle increment--set CHAR+5 to 21030060(8)
  convert standard setting to character
SCAN:
  reset interface
  if φ backlashes
    then subtract .3 degrees from φ angle
  convert angle to character
  if Ψ backlashes
    then subtract .3 degrees for Ψ angle
       convert angle to character
  if either angle is backlashed
    then do
      set count against time
      load time register with one count
    end
    else do
      if Ψ moves
        then do
PHIPSI:
          if φ moves
            then update φ angle
PSI:
          update Ψ angle
        end
        else do
          convert number of steps to character
          convert angle increment to character
        end
SETCT:
      if count interval <0 --PRESET
        then set count against time
        else set count against monitor
      load negative interval into hold register
      read data register to load hold register into
         proper count register
    end
```

```
LOAD:
  set interrupt counter--set TEMP+4 -6
ℓ1:
  set character/word counter--set SAVE+1 2
  set number of words to be sent--set SAVE -8
  store branch CHRLD in location ADRES
CHRLD:
  if all words sent
    then do
ENDLD:
      disable character interrupt
      send end of block character to Mitsubishi
    end
    else do
      load word being processed
      set bit for character to be sent
      send the character
      if all characters in word sent
        then do
          increment the word counter
          reset character counter--set SAVE+1 2
        end
    end
  return to system

DATAIN:
  if number of steps in scan is 0
    then disable interrupt
  if Ψ moves
    then do
      enable the character interrupt
      store branch SCAN in location ADRES
    end
  if count against monitor
    then load contents of time register
    else load contents of monitor register
  convert non-preset to 15 float
  convert data to 15 float
  send data interrupt to 15
  if Ψ moves
    then if Ψ increment < 0
      then reset Ψ backlash bit
      else if φ increment < 0
        then reset φ backlash bit
  return to system
```

```
UNDCHR:
  if 6 attempts have been made to send block
    then do
CHREND:
      send end of tape character to Mitsubishi
      send undefined character interrupt to
           15--ONCODE=4
      branch to initialize the Mitsubishi
    end
    else do
      clear Mitsubishi
      increment interrupt counter
      branch to character send code--label ℓ1
    end

SNDATA:
  send data to 15
  if number of steps remaining > 0
    then return to system
    else do
      enable character interrupt
      load THUMB re-entry address
      send interrupt to 15 for new scan--ONCODE=1
    end

CLKB:
  if number of clock ticks is 0
    then schedule THUMB for execution
    else do
      increment clock tick counter
      schedule medium clock
    end

CHINT:
  if interrupt enabled
    then schedule THUMB for execution
  return to system

DTINT:
  if character interrupt enabled
    then store branch UNDCHR in location ADRES
    else
      if interrupt EOM
        then do
EOM:
          if either backlash bit set
            then do
```

```
BCKLSH:
              reset backlash bit
              store branch START in location ADRES
              read data register
              enable character interrupt
              return to system
            end
            else
        else do
EOT:
          clear Mitsubishi
          set up interrupt oncode for 15--ONCODE=8
        end

TASK:
  store branch INT in location ADRES
  schedule THUMB for execution
  return to system

SETUP:
  store branch INIT in location ADRES
  schedule THUMB for execution

SHTDWN:
  send end of tape character to Mitsubishi
  return to system
```

Variable term dictionary for Thumb and Finger.

```
DATIN:
  +0:  GA-arm angle setting (φ in comments)
  +1:  ST-arm angle setting (Ψ in comments)
  +2:  increment for angle φ
  +3:  number of steps in scan
  +4:  standard setting
  +6:  move word
         bit 23 set if φ moves
         bit 22 set if Ψ moves
         bit 21 set if φ backlashed
         bit 20 set if Ψ backlashed
  +7:  counting interval

CHR:
  +0:  string for zero stepsize
  +1:  string for one step scan
CHAR:  array containing the converted output for the
           Mitsubishi
INTR:  location false for ignoring character interrupt,
           true for processing it
```

C. Triple Axis Codes

Unlike the Mitsubishi diffractometers, the Triple Axis diffractometer has no computer of its own for angle driving etc. Thus, all of the operations are controlled directly by the 910. This control is exercised through a number of registers in the interface for loading values and a control register for handling hardware operations.

The angle setting for each arm of the Triple Axis is measured by a 16 bit binary encoder. This means that the decimal angle values from 0 to 36000 in hundreths of a degree are represented by only 32768 distinct settings. Hence, conversion from octal to decimal must be made both in the hardware decimal display and the software drive routine. Note that all angle values cannot be represented on the display panel since there are only 32768 possible settings. In order to drive a given arm as close to its true value as possible, the decimal angle value is converted to octal. This conversion yields an irrational number in general. The arm is driven to within .5 degrees of the desired angle when starting a scan and for all negative angle increments to remove backlash from the gears. The arm is then stepped to within .03 degrees of the desired setting. At this point each arm is stepped one motor pulse at a time until the desired encoder position is reached. The fractional part of the octal angle value is then converted to motor steps. The arm is further stepped until this number of steps counts down to zero.

The single pulsing of the motors is done to avoid overshooting the desired arm setting. If an overshoot should occur, the arm driving procedure is repeated for that arm. After three such overshoots, the arm is left at its position and an interrupt with oncode 64 plus the arm pointer for the arm causing the problem is sent to the 15.

When the 15 sends a scan to be performed, entry point TRIAX is scheduled by the 910 system. The angles are converted to octal and the initial number of motor steps to drive each arm is calculated and sent to the interface. Note that the register cannot be loaded with zero motor steps as a motor stop interrupt will never occur. When all initial increments have been loaded, all the motors are started by setting appropriate bits in the control address register.

When each arm reaches its position, a motor stop pulse is generated. This causes an interrupt to be sent to the 910. The interrupt handler reads the status register to determine which arm caused the interrupt. Entry point TRIAX is then scheduled by the 910 system for the next movement of the arm. The arms are now processed one at a time until they reach their proper positions for the count. The respective bit in the DONE word is then set. When all nine bits of the DONE word are set, the count is started. Upon completion of the count, an interrupt is received in the 910. The 910 then reads the data and background registers, and signals the 15 that the count is complete. The 15 then requests that the data be transmitted. If the scan is completed, then no new task is started. Otherwise, the 910 drives the arms to the next position in the scan and starts another count.

The data registers are 18 bit registers which only store positive values. They are loaded into the 18 high order bits of the 910. The registers can be scaled to expand their range. This is done by setting the data to indicate 1 count for every 2 counts etc. When the data are loaded into the 910, the high order 18 bits contain the contents of the 18 bit count register and the low order 6 bits contain the prescale bit. Thus, bit 23 is tested. If it is set, then the bits to the left contain the data point. If it is reset, the register is shifted right 1 bit and bit 23 is tested again. This process is repeated until bit 23 is set. One more right shift retrieves the data.

The Triple Axis has special interrupts for processing an abnormal condition. These conditions are high and low limit switches on the arms themselves, along with a bumper switch which is activated if two arms collide. Finally, the interface has a dump button which stops the diffractometer no matter what it is doing. All interrupts sent to the 15 require an oncode. For abnormal interrupts, the angle generating the interrupt needs to be known by the 15. This is done by constructing an oncode with bits 9/11 set for the cause of the interrupt, and bits 12/17 containing the respective angle pointer. The angles are numbered from θ_M to θ_{A2} in order.

Priority	Oncode	Interrupt
1	0	panic stop
30	1	count complete
21	2	θ_M step complete
22	3	$2\theta_M$ step complete
23	4	Ψ step complete
24	5	$2\theta_{S1}$ step complete
25	6	θ_{A1} step complete
26	7	$2\theta_{A1}$ step complete
27	8	$2\theta_{S2}$ step complete
28	9	θ_{A2} step complete
29	10	$2\theta_{A2}$ step complete
03	11	θ_M low limit switch
05	12	$2\theta_M$ low limit switch
07	13	Ψ low limit switch
09	14	$2\theta_{S1}$ low limit switch
11	15	θ_{A1} low limit switch
13	16	$2\theta_{A1}$ low limit switch
15	17	$2\theta_{S2}$ low limit switch
17	18	θ_{A2} low limit switch
19	19	$2\theta_{A2}$ low limit switch
4	20	θ_M high limit switch
6	21	$2\theta_M$ high limit switch
8	22	Ψ high limit switch
10	23	$2\theta_{S1}$ high limit switch
12	24	θ_{A1} high limit switch
14	25	$2\theta_{A1}$ high limit switch
16	26	$2\theta_{S2}$ high limit switch
18	27	θ_{A2} high limit switch
20	28	$2\theta_{A2}$ high limit switch
2	29	bumper limit switch

Oncode Bit	Interrupt
17	count complete
16	panic stop
15	bumper limit
11	overshoot
10	low limit switch
9	high limit switch

Table 2. Interrupt status register and 15 interrupt oncodes.

Bit	Function
0	
1	
2	
3	stop oscillator
4	start oscillator
5	unused
6	set wait (time preset)
7	set normal (monitor preset)
8	start count
9	stop count
10	unused
11	start θ_M motor
12	start $2\theta_M$ motor
13	start Ψ motor
14	start $2\theta_{S1}$ motor
15	start θ_{A1} motor
16	start $2\theta_{A1}$ motor
17	start $2\theta_{S2}$ motor
18	start θ_{A2} motor
19	start $2\theta_{A2}$ motor
20	interrupt flag clear and motor stop
21	read done (reload interrupt register)
22	clear interrupt process
23	set interrupt process (block interrupt)

Table 3. Control address register.

Address	Output Function
230(8)	load θ_M step register
231(8)	load $2\theta_M$ step register
232(8)	load Ψ step register
233(8)	load $2\theta_{S1}$ step register
234(8)	load θ_{A1} step register
235(8)	load $2\theta_{A1}$ step register
236(8)	load $2\theta_{S2}$ step register
237(8)	load θ_{A2} step register
240(8)	load $2\theta_{A2}$ step register
241(8)	load monitor preset register
247(8)	control address register

Address	Input Function
210(8)	interrupt status register
211(8)	θ_M encoder
212(8)	$2\theta_M$ encoder
213(8)	Ψ encoder
214(8)	$2\theta_{S1}$ encoder
215(8)	θ_{A1} encoder
216(8)	$2\theta_{A1}$ encoder
217(8)	$2\theta_{S2}$ encoder
220(8)	θ_{A2} encoder
221(8)	$2\theta_{S2}$ encoder
222(8)	data 1 input register
223(8)	background 1 input register
224(8)	data 2 input register
225(8)	background 2 register
226(8)	counting time register
227(8)	counting monitor register

Table 4. Input/output registers.

The following entry points and buffer areas exist for the 910 Triple Axis code.

DRINT: interrupt handler

STRTUP: entry point for OPEN (TRIAX)

SHTDWN: entry point for CLOSE (TRIAX)

TRIAX: entry point for scan processing from WRITE (TRIAX) KEYFROM(1). ANGLE contains the scan data written from the 15.

SNDATA: entry point for data transmission from READ TRIAX KEY(2). DATA contains the count data being sent to the 15.

ENCDRD: entry point for encoder positions from READ TRIAX KEY(3). ENCOD contains the encoder positions to be transferred to the 15.

REDSET: entry point for pre-scale multiplier read from READ (TRIAX) KEY(4). ENCOD contains the multiplier to be sent to the 15.

```
TRIAX:
  branch to proper entry point--initially BEGIN
BEGIN:
  set bits in DONE for angles which will not backlash
  do until all angles are processed
    if angle is to be moved
      then if angle is not to be backlashed
        then reset corresponding bit in DONE word
  end
  if counting against time--PRESET <0
    then set monitor bit in control address
    else set time bit in control address
  output stop motor and clear flag to Triple Axis
  output preset value to control address
```

```
DRIVER:
  load branch RSTRT into location BRNCH
  reset all bits in motor start word
  load re-entry point
  if all angles properly set
    then do
 COUNT:
      output counting interval to interface
      start the count
      reset bits in DONE word for arms which move
  EXIT:
      return to system
    end
    else do
      initialize overshoot count--set TRY 2
      set final test branch array--set PH TEST
 LOOP1:
      do until all angles are checked
        if angle is to be moved
          then do
            convert angle to encoder position
            convert encoder position to fixed point
            store the integral encoder value
            store the fractional encoder value
          end
    end
   TOP:
     do until all angles are processed
       save the angle pointer
       if angle is to be moved
         then do
 RSTRT:
           reset motor start bits
           read arm position
           set sign word positive
           if desired angle <4
             then if encoder > 32764
               then add 32768 to encoder position
           subtract present encoder position
           if increment is negative
             then do
               mark sign word negative
               change increment positive
             end
           convert encoder position to float
           convert to motor steps
           convert motor steps to fixed
           if angle is θM, Ψ, θA1, θA2
```

```
                    then do
                      if increment > 180 degrees
                        then do
                          convert increment to compliment
                          change the sign word
                        end
                    end
        CONT1:
                  if arm is to be backlashed
                    then do
        CONT2:
                      increment=increment-.5 degrees
                      if 3 overshoots have occurred
                        then do
                          set up oncode for 15
                          output interrupt to the 15
                          branch to DNE
                        end
                      decrement overshoot counter
                      if number of motor steps is negative
                        then set bit 0 to mark negative move
                    end
                    else do
                      if number of motor steps <=1
                        then
                          if number of motor steps 0
                            then do
                              branch to TEST or LAST
        TEST:
                              load fractional encoder position
                              normalize the mantissa
                              convert to motor steps
                              convert to fixed
                              set branch point
                              decrement the last output array
        LAST:
                              load one step for sending
                            end
                        else do
                          if increment negative
                            then if initial arm drive
                              then branch to CONT2
                              else do
        DNE:
                                set proper bit in DONE
                                if angle not scanned
                                  then set proper backlash bit
```

```
                        if all bits set in DONE word
                          then branch to COUNT
                          else branch to EXIT
                    end
                  branch to 125
            end
            else do
CONT3:
              if increment is negative
                then do
  ℓ25:
                  reload test branch for overshoot
                  branch to CONT2
                end
              if number of motor steps > .03 degrees
                then advance to angle less .03 degrees
                else advance one motor step
            end
          end
 LOAD:
          if number of motor steps 0
            then change to .5 degrees
          output number of motor steps
          set proper bit in START word
          if not first time through loop
            then go to STRT
        end
ENDLP:
  output START word to interface
 STRT:
  branch to EXIT

 SCAN:
LOOP3:
  initialize pointers
  do until all angles have been processed
    if increment pointer is 0
      then reset increment pointer
    do until all angles are processed
      if angle moves then go to ℓ3
    end
    branch to DRIVER
ℓ3:
  end
  add angle increment to current value
  branch to DRIVER
```

```
DATAIN:
  if counted with detector 1
    then do
      call INPUT to process counts on 1
      call INPUT to process background on 1
    end
DTWO:
  if counted with detector 2
    then do
      call INPUT to process counts on 2
      call INPUT to process background on 2
    end
CTINT:
  call INPUT to process non-preset count interval
  return to system

INPUT:
  do while bit 23 is not set
    right shift register 1 bit
  end
  convert to 15 floating point representation
  if data value is negative
    then do
      reset sign bit
      increment exponent--multiply by 2
    end
  exit subroutine

ENCDED:
  read all encoder positions
  output values to 15
  return to system

SNDATA:
  transmit the data to the 15
  decrement the number of steps
  if number of steps is 0
    then load branch BEGIN into BRNCH
    else do
      if primary increment is 0
        then load branch COUNT into BRNCH
        else load branch SCAN into BRNCH
      schedule TRIAX
    end
  return to system
```

```
RDSET:
  read the prescale multiplier value
  convert to 15 float representation
  transmit value to 15
  return to system

STRTUP:
  load branch BEGIN into BRNCH
  return to system

DRINT:
  read the status register for oncode
  if oncode is 0
    then panic interrupt
    else if oncode is 1
      then load branch DATAIN into BRNCH
      else if oncode <11
        then do
          mark angle pointer
          schedule next arm drive
        end
       else if oncode < 20
         then low limit switch interrupt
         else bumper limit interrupt
  return to system
```

Variable term dictionary for TRIAX.

POS: array of input EOM commands for the encodes.

STEP: array of output EOM commands for motors.

STEP1: array of number of motor steps/encoder mark.

HDEG: array of number of motor steps/half degree.

THDEG: array of number of motor steps/.03 degrees.

MSTP: array of mantissas for number of motor steps/encoder mark.

MEXP: array of exponents for number of motor steps/encoder mark.

ADJST: round off factor for float-fixed conversion.

START: bits 11/19 are set for starting respective motors for a given angle. The word is sent to the interface through the control address register.

DONE: bits 15/23 are set when a respective arm is at its desired position. Otherwise the bits are reset.

PT: pointer for the arm being processed. This value is loaded into the index register for an indexed load of the various arrays.

F: word set false for the initial drive of the arms and true thereafter. Initially all angles which are to be driven are started simultaneously. The false value of F prevents exit from the initial driving loop until all arm movements are calculated. Thereafter, arms are processed one at a time as their interrupts occur. The true value of F exits the driving loop after one pass through.

TEMP: temporary storage.

DELTA: temporary storage of angle increment.

SIGN: location to mark sign of angle increment. False if positive increment, and true if negative increment.

ANGLE: array of the angle values in hundreths of degree.

BCKLSH: bits 15/23 are set if arm is not to be moved at any time during the scan. After the initial backlash, all bits are set for all angles which do not change during the scan.

MOVE: bits 15/23 respectively are set if corresponding arm moves during the scan. Bits 13/14 are set for counting on detectors 2 and/or 1 respectively.

PRESET: the present count interval. This value is negative for counting against the clock, and positive for counting against the monitor.

AINC: angle increments.

NSTEP: number of steps in the scan.

PH: branches for final arm positioning. Originally converts fraction encoder position to motor steps. Then outputs final motor pulses one at a time.

ENCINT: array of the integral number of encoder positions for the proper angle setting. Drives the arm to the proper integral encoder position.

ENCFRA: array of fractional number of encoder positions for the proper angle setting. Used for final angle driving.

LSTEP: array of conversion from fractional encoder positions to motor steps for final angle driving.

TRY: array of counters for the number of overshoot attempts. Initially set to 2.

D. Utility Programs for 910 Codes

There are two utility tasks used in conjunction with the 910 neutron diffraction codes: a code for conversion from fixed to character, and a code for conversion from 910 floating point representation to 15 floating point representation. These routines will be discussed below.

1. C2-the fixed to character code. This code converts fixed numbers to ASKII characters to drive the Mitsubishi. The letters are represented by the 8 bit octal numbers 101(8)-132(8) and the digits by the 8 bit octal numbers 060(8)-071(8). The two special characters required are 15(8) for the carriage return and 72(8) for the semi colon. The final character string consists of a letter followed by 2 or 5 digits. The letters are not changed. The 2 digits following a letter are converted and stored in the same word as the letter. If there are 5 digits following a given letter, a second pass is made through the conversion loop. Three characters are packed into the word following the letter.

Upon entry to this routine the A register contains the decimal number to be converted, the B register contains the power of ten of this number in scientific notation, and the X register contains the address at which the resultant character string will be stored. There is nothing returned from this routine.

The characters are packed, three per word from left to right. When they are sent, the eighth bit in the character being sent is set.

The algorithm is as follows:

```
    save the parameters passed from the calling routine
    strip old digit characters from character string
    set character index--set TEMP+3
  LOOP2:
    do while character index >0
      divide number by proper ten to proper power to
        get digit
      convert the digit to character--add 60(8)
      shift the character to its proper location
      merge with character string
      multiply extracted digit by ten to proper power
      subtract from starting number
      decrement power of ten
      if power of ten <0
        then exit routine
    end
    increment character string address
    reset character index--set TEMP+3 2
```

2. <u>CNVRT-the 910 float to 15 float code</u>. This code converts SDS-910 24 bit floating point numbers to PDP-15 18 bit floating point numbers. The number to be converted is passed in the A and B registers. The converted number is returned to the calling routine in the A and B registers. The algorithm is as follows:

```
    save the mantissa
    remove all bits but the exponent bits from exponent
    right shift mantissa 6 bits
    save the corrected mantissa
    right shift mantissa bits in B register 6 bits
    merge the exponent with low order mantissa bits
    return mantissa to B register for return
    exit routine
```

V. DATA SET ORGANIZATION

Each experiment library has two primary data sets and two secondary data sets. The primary data sets are SCANIN, the data set containing the scan files, and DATAOUT, the data set containing the count files. The secondary data sets are SCANTEST which contains scan files for running task TESTQE simultaneously with QSCAN, and ANGSTR which contains the angles computed in the calculation routine for q-scans.

A. Triple Axis Data Sets

SCANIN: the files for data set scanin are set up as follows:

FILE 0:

word 1: experiment identifier for restart purposes. It contains the characters GEN, DBL, or QES depending on the experiment being performed.
word 2: contains the bits set for fixing angles. Bits 9/17 are set if θ_M-$2\theta_{A2}$ are never to move during a scan.
word 3: input title which is printed as heading for each scan.

FILE 1: This file contains half of the semi-permanent data which are stored as follows:

word 1: ST1, sample table angle of calibration 1.
word 2: SCAN, contains the scan indices for transforming from 3-d space to the plane of the scan:
bits 15-17 contain index of component 1 of the 2-d $\vec{q}$.
bits 12-14 contain index of component 2 of the 2-d $\vec{q}$.

bits 9--11 contain index of component 3 of the 2-d $\vec{q}$.
bit 8 is set for an HKH or HHL plane.
bit 7 is set for an HKK plane.

word 3: TAG, 3 characters specifying the plane of the 2-d $\vec{q}$ in 3 space.
word 4: PSI1, $ST1+\pi/2-(\varphi_1 - \varphi_2)/2$.
word 5: PSI2, sample table angle for calibration 2.
word 6: PHI1, the $2\theta_{S1}$ angle for calibration 1.
word 7: PSI2, the $2\theta_{S2}$ angle for calibration 2.
word 8: PHI0, $(\varphi_1-\varphi_2)/2$.
word 9: PSI0, $\pi/2$-PHI0.
word 10: DA1, d-spacing for the crystal in the analyzer for detector 1.
word 12: DA2, d-spacing of the analyser crystal for detector 2.
word 14: DM, d-spacing of the monochromator crystal.
word 16: DEN, the denominator for a and b.
word 17: ETA, the angle between the calibration vectors $\vec{G}_1$ and $\vec{G}_2$.
word 18: ANG0, the zero reference angles in order $2\theta_{A2}$ to θ_M.

FILE 2: This file contains half of the semi-permanent data which are stored as follows:

word 1: ANGHI, the high limit angles for the angle calculation in order $2\theta_{A2}$ to θ_M.
word 8: ANGLO, the low limit angles for the angle calculation in order $2\theta_{A2}$ to θ_M.
word 15: E11, component 1 of calibration vector $\vec{G}_1$ in 2-space.
word 16: E12, component 2 of calibration vector $\vec{G}_1$ in 2-space.
word 17: E21, component 1 of calibration vector $\vec{G}_2$ in 2-space.
word 18: E22, component 2 of calibration vector $\vec{G}_2$ in 2-space.
word 19: G_1, magnitude of calibration vector 1.
word 21: G_2, magnitude of calibration vector 2.
word 23: LAMDA, calibration wavelength.
word 25: SIGN, bits 13/17 are set for right hand rotation from reference line for angles $2\theta_M, 2\theta_{S1}, 2\theta_{A1}, 2\theta_{S2}, 2\theta_{A2}$ respectively. Bits are reset for left hand rotation.

FILES 3/19: These files contain the scan data for each of the scans to be run. The pertinent scan information will be listed for each experiment separately.

1. GENSCAN:

word 1: NSTEP, bits 0/8 contain the number of steps to be performed in the scan.
bits 9/17 contain the number of steps completed in the scan.
The scan is considered complete if the two values are equal.

word 2: C, bit 17 set for counting on detector 1.
bit 16 set for counting on detector 2.

word 3: SCNTYP, bits 9/17 are set if corresponding angles θ_M to $2\theta_{A2}$ are to be moved during the scan.
bit 8 set for counting on detector 1.
bit 7 set for counting on detector 2.

word 4: EOF, bit 13 set if counting done on both detectors.
bit 12 set for last scan on data set.
bits 0/8 contain the file number for the count data.

word 5: PT1, 1-scan moves two angles, 1 detector.
2-scan moves four angles, 2 detectors.
3-scan moves one angle, 1 detector.
4-scan moves two angles, 2 detectors.

word 6: BCKLSH, bits 9/17 contain bits set for angles not backlashed during the scan. Bits are reset for backlash.

word 7: OUTKEY, contains the key for the output record on the count data set. Originally set to starting record of first scan.

word 8: PRESET, the counting interval, <0 if counting against time, >0 if counting against monitor.

word 9: PT, array containing the indices of the angles which move during the scan. Used for coordinating teletype output.

word 14: DUM, unused dummy parameter.

word 15: STEP, array containing the angle increments of the angles being scanned.

word 17: ANG, array containing the initial angles of the scan.

2. DBLRCK:

word 1: NSTEP, bits 0/8 contain number of steps each side center angle of second scan.
bits 9/17 contain number of steps each side center angle for first scan.

word 2: FILESTRT, contains the starting key of the count data set file. Initially set to the starting key for the first scan.

word 3: SCAN, bits 9/17 are set for corresponding angles θ_M to $2\theta_{A2}$ if angles are to move during scan.
bit 8 set for counting on detector 1.
bit 7 set for counting on detector 2.
bit 5 set if scan 2 coupled.

word 4: EOF, bit 14 set if scan complete.
bit 12 set for the last scan on data set.
bit 15 set if scan in progress.
bit 16/17 contain iteration counter. It is incremented whenever the change in the center angle is less than the convergence criterion for that angle. It is set to zero whenever the above is not valid. The sample is considered centered when the counter reaches 2.
bits 0/8 contain the count data set file.

word 5: NOS, bits 9/17 contain index of angle 1.
bits 0/8 contain index of scan angle 2.

word 6: NO, contains the key of the current record being used on the count data set.

word 7: OUTKEY, contains the key of the current record being used on the count data set.

word 8: MULT, contains the multiplier factor for scan if quickscan. The stepsizes are multiplied by this value if the scan seems too narrow.
ANG1,ANG2, after start of scan, ANG1 is the current value of the first scan angle and ANG2 is the current value of the second scan angle.

word 10: START, set to value of current scan number in the double rock pair.

word 11: CNV1, convergence criterion for scan 1.

word 12: CNV2, convergence criterion for scan 2.

word 13: BCKLSH, bits 9/17 set for corresponding angles θ_M to $2\theta_{A2}$ not to be backlashed.

word 14: PRESET, counting interval of scan: <0 for counting against time. >0 for counting against monitor.
0 for performing quickscan at every other point to determine counting interval such that maximum counts of spectrum is 2000.

word 15: STEP1, step size for scan 1 of pair.

word 16: STEP2, step size for scan 2 of pair.

word 17: ANG, array containing the initial angles of the scan.

3. QESCAN:

word 1: NSTEPS, bits 0-8 contain the number of steps in the scan.
bits 9-17 contain the number of steps completed. These numbers are equal when the scan is complete.

word 2: SCAN, see word 2 of file 1.

word 3: TAG, see word 3 of file 1.

word 4: PRIN_EN, bit 17 is set for counting on detector 1.
bit 16 is set for counting on detector 2.
bit 15 is set for fixed incident energy.
bit 14 is set for fixed scattered energy.
bit 13 is set of both detectors are used.
bit 12 is set for last file on data set.
bits 0/8 contain the file number for the data on the count data set.

word 5: NC, bits 0/8 contain the number of counts taken at each angle setting.

word 6: PRESET, count interval: <0 for count against time, >0 for count against monitor.

word 7: KEY1, current key for count data set. Initially set to starting key for first scan on the file.

word 8: QP1, component 1 of principal $\vec{q}$.

word 10: DQ1, component 1 of the $\vec{q}$ increment.

word 12: QP2, component 2 of principal $\vec{q}$.

word 14: DQ2, component 2 of the $\vec{q}$ increment,

word 16: NU, value of the inelastic phonon energy.

word 17: DNU, increment of inelastic phonon energy.

word 18: ENERGY, energy value of the fixed energy.

word 22: QW1, component 1 of the weak $\vec{q}$.

word 24: QW2, component 2 of the weak $\vec{q}$.

SCANTEST: This data set is used when a dummy angle calculation is desired. Two copies of the semi-permanent data are maintained on this file: one identical to the current semi-permanent data, and one for updating for a given test run.

FILE 0: This file is not used.

FILES 1/2: Semi-permanent data used by TESTQE. The current semi-permanent data are copied at the beginning of a run. Hence changes made to the semi-permanent data during test are not entered onto the disk.

FILES 3/4: Copy of the semi-permanent data changed only when semi-permanent data changed in file SCANIN using task START. DUMPSPD dumps this copy of the semi-permanent data when it is invoked.

FILES 5/8: Scan files for test of q-e scan angles.

DATAOUT: This data set consists of variable length files consisting of two word records. The files contain the following information:

FILE 0: Six character string TRIAX to mark the directory of the data set. This allows tasks which can access this data set to determine which DATAOUT data set they are processing.

FILE 1: Character string of 11 records containing the alphanumerics used as the scan title for printing on the teletype.

FILE 2: Twenty records containing the partial counts for multiple counts at one angle setting. The first ten records are for detector 1, and the second ten records are for detector 2.

When counting is complete, these values are summed and entered into the count FILES.

FILES 3/N: These are the count files which contain the initial scan values as well as the total counts at each angle setting. These files are variable length. Each file has a pointer which marks the starting key of the next file on the data set. These files differ slightly depending upon the experiment being performed, but different types of count files may be stored simultaneously. They are described separately below:

1. Genscan:

```
record 0:
  word 1:  number of detectors in use.
  word 2:  pointer for starting key on next count file.
record 1:
  word 1:  initial value of scan angle.
  word 2:  increment of scan angle.
             These angle increments are entered in order
             from θM to 2θA2, but are not specifically
             marked. Only values are entered which are
             scanned.
record 2:  the number of counts at each angle
             setting are stored in floating point words,
             one value per record.
```

2. Double Rock:

```
record 0:
  word 1:  pointer to key for start of second
             spectrum on the file.
  word 2:  pointer to key for start of second
             spectrum of the file.
record 1:
  word 1:  initial angle of scan 1.
  word 2:  angle increment of scan 1.
record 2:  the number of counts at each
             angle setting.
record n+1:  initial angle of scan 2.
  word 1:  initial angle of scan 2.
  word 2:  angle increment of scan 2.
record n+2:  the number of counts at each angle
             setting.
```

<u>Q(e)scan</u>:

record 0:
word 1: the number of detectors stored as a negative fixed number.
word 2: the pointer to the key for the start of the next file on the data set.
record 1: components of the principal q.
record 4: value of the phonon scattering energy.
record 5: components of the weak $\vec{q}$.
record 6: increments of $\vec{q}$.
record 11: increment of the phonon energy.
record 12: fixed beam energy.
record 13: the number of counts at each angle setting.

ANGSTR: This data set consists of 2 files, one of 15 records and one of 8 records. Each record is 11 words long. The first 9 words contain the angles of the scan, and the last 2 words contain the floating point value of the phonon energy for the second detector. The first file is used by the scan task, and the second file is used by the test task so that they can be executed simultaneously.

B. Finger and Thumb Data Sets

These data sets are set up similarly to the Triple Axis data sets. Where the files are identical, reference will be made to the corresponding identification in the above data sets. The Finger and Thumb data sets are almost identical in makeup. Only the Thumb data sets will be listed with the exception of when the Finger data sets differ. In this case the Finger file will be given below the Thumb file.

SCANIN:

FILE 0: First half of teletype heading.

FILE 1: Second half of teletype heading.

FILE 2:

word 1: H1, h-index for calibration one.
word 2: K1, k-index for calibration two.
word 3: L1, ℓ-index for calibration one.
word 4: H2, h-index for calibration two.
word 5: K2, k-index for calibration two
word 6: L2, ℓ-index for calibration two.
word 7: ST1, sample table setting for calibration one.
word 8: ST2, sample table setting for calibration two.
word 9: ETA, angle between the calibration vectors.

FILE 3:

word 1: GA1, GA-arm setting for calibration one.
word 2: GA2, GA-arm setting for calibration two.
word 3: STLO, low limit angle for sample table.
word 4: STHI, high limit angle for sample table.
word 5: GALO, low limit angle for GA-arm.
word 6: GAHI, high limit angle for GA-arm.
word 7: VALO, low limit angle for VA-arm.
word 8: VAHI, high limit angle for VA-arm.
word 9: SIGN, sign bits for GA and VA arms.
Words 7/9 are unused in Thumb.

FILE 4:

word 1: LAMCALC, the calibration wavelength.
word 2: TAG, plane of the scan.
word 3: GAZ, zero reference angle for GA-arm.
word 4: KO, calibration momentum value.
word 7: SNGA, sign bit for GA-arm.
words 8/9: unused in Thumb.

word 1: LAMCALC, calibration wavelength.
word 3: TAG.
word 4: KO.
word 6: GAZ.
word 7: VAZ, zero reference angle for VA-arm.
word 8: DA, d-spacing of analyzer crystal.

FILE 5:

word 1: G_1, magnitude of calibration vector one.
word 3: G_2, magnitude of calibration vector two.
word 5: D, denominator of plane.
word 5: E11, unit vector one for calibration one.
word 6: E12, unit vector two for calibration one.
word 7: E21, unit vector one for calibration two.
word 9: E22, unit vector two for calibration two.

FILES 6/N:

1. Genscan:

word 1: NSL, number of completed steps in scan.
word 2: NS, total number of steps in the scan.
Scan is marked complete when NS=NSL.
word 3: SCNT, bit 17 is set for coupled GA scan;
bit 16 is set for uncoupled GA scan;
bit 15 is set for sample table scan.
word 4: DEL, angle increment.
word 5: PRESET, counting interval.
word 6: ST, current sample table angle setting.
word 7: GA, current GA-arm angle setting.
word 8: OUTKEY, key for count data set.
word 9: EOF, bits 0/12 contain count file number;
bit 17 is set for last scan file.

word 1: NS, bits 0/8 contain total number of
steps in scan; bits 9/17 contain number
of steps completed in scan.
word 2: SCNT.
word 3: DEL.
word 4: PRESET.
word 5: ST.
word 6: GA.
word 7: VA, current angle setting of VA-arm.
word 8: OUTKEY.
word 9: EOF.

2. Double Rock: Each scan in the double rock experiments requires two records of the scan file.

FILE 1:

word 1: KT, convergence counter.
word 2: NS, number of steps left in scan.
word 3: STYP, bit 17 set if uncoupled GA scan, reset if coupled GA scan; bit 16 set if current scan is GA-arm, reset if current scan is sample table scan; bit 15 is set if scan is in progress; bit 14 is set if scan is completed.
word 4: ST, current value of sample table angle.
word 5: GA, current value of GA-arm setting.
word 6: STC, center angle for sample table scan.
word 7: GAC, center angle for GA-arm scan.
word 8: OUTKEY.
word 9: EOF.

word 1: NSTEP, bits 0/8 contain number of steps for sample table scan; bits 9/17 contain number of steps for GA-arm scan.
word 2: STRT, starting key of count file for scan.
word 3: MULT, multiplier for quickscan.
word 5: DELST, angle increment for sample table scan.
word 6: CON1, convergence criterion for sample table.
word 7: DELGA, angle increment for GA-arm scan.
word 8: CON2, convergence criterion for GA-arm.
word 9: PRESET.

FINGER:

word 1: NS.
word 2: STYP, as above; bits 7/8 contain iteration counter.
word 3: ST.
word 4: GA.
word 5: VA, current value of VA-arm
word 6: STC.
word 7: GAC.
word 8: OUTKEY.
word 9: EOF.

3. Qescan: The files in the qescan experiments also require two records per scan.

word 1: NSL.
word 2: NS.
word 3: TAG.
word 4: NC, number of counts at one setting.

word 5: K0, calibration momentum magnitude.
word 7: NH, counts left to perform at one setting.
word 8: KEY1, current value of count data set key.
word 9: EOF.

word 1: PREST, counting interval.
word 2: Q1, component one of $\vec{q}$ of scan.
word 4: Q2, component two of $\vec{q}$ of scan.
word 6: DQ1, component one of $\vec{q}$ increment.
word 8: DQ2, component two of $\vec{q}$ increment.

FINGER:

word 1: NS.
word 2: NC, bits 0/8 contain number of counts per angle setting; bits 9/17 contain number of counts performed at that setting
word 3: TAG.
word 4: NU, phonon energy of inelastic scattering.
word 6: DNU, phonon energy increment.
word 7: KEY1.
word 8: EOF.

VI. ACKNOWLEDGEMENTS

This work was supported by the U.S. Department of Energy, Division of Basic Energy Sciences.

The authors wish to thank L. C. Moon for valuable discussions and input into the development of this system as well as assistance with some of the implementation.

Special thanks are due to W. Lembke for her expert typing as well as her handling of the details of getting this manual printed.