technical memorandum Daresbury Laborator

DL/SCI/TM80E

SPECIFICATION FOR SRS WIGGLER-2 STATION 16.3 DATA ACQUISITION SYSTEM

by

M.C. MILLER and S. AHERN, SERC Daresbury Laboratory.

JULY, 1991

91/272

Science and Engineering Research Council

DARESBURY LABORATORY

Daresbury, Warrington WA4 4AD

ISSN 0144-5677

Specification for SRS Wiggler-2 Station 16.3 Data Acquisition System

M.C.Miller and S.Ahern

24th June 1991

Contents

Part I

HARDWARE

1 Introduction

This document is designed to itemise and describe the data acquisition requirements for both the hardware and software to be provided for the 2nd Wiggler Station 16.3. It was derived after meetings involving R.Cernik, G.Oszlanyi, M.C.Miller, and S.Ahern. The station is to be an X-ray diffraction station for both high resolution studies, crystallographic applications and probably also magnetic diffraction. It will be fairly similar in operation to the existing 9.4 station. This opportunity is being taken to completely redesign the next generation of SRS software using high performance hardware and the latest methods. The program must be modular and extensible both in terms of functions provided for the user and internal structure as programmed. There is expected to be a considerable benefit from this work for existing stations and elsewhere. The station is expected to be fully operational in 1993.

2 Experimental Methods

A monochromatic beam of X-rays is required which can be sagitally focussed to increase the flux for weak scattering applications. In any case, at high energies of say 80KeV there should be a three-fold increase in the flux compared to 9.4. In addition, it is possible that white beam may be allowed into the hutch. The station requires a medium or heavy duty diffractometer for use with single crystal samples. It must be optimised for materials studies in the field of diffuse scattering, phase transitions, anomalous dispersion, weak scatterers, line shape analysis, the effects of extinctions and magnetic diffraction. It must accommodate large environment cells such as furnaces, cryostats, pressure cells, magnets etc. There must also be the ability to run the diffractometer in triple-axis mode.

3 Station Layout

Station 16.3 is the end-of-line station on the 2nd Wiggler. As such, the experimental area will be about 30m from the source in the old NINA tunnel. Here will be situated the hutch, diffractometer and adjacent control area containing data acquisition and control electronics, computer racks and terminals. The main monochrometer is housed separately in the line 16 optics area about 15m from the source. Some instrumentation in the optics area will therefore be controlled from 16.3 over a distance of about 21m and channels are planned to carry any necessary cabling safely between them.

4 Hardware Functions Required

The following items are the most likely to appear at the current stage of thinking but one or two of the minor movements may be varied later with no great effect on the data acquisition design.

4.1 Optics Area

- High resolution Theta-rotation drive for specifying wavelength (double crystal mono).
- Drive to bend top mono crystal.
- Drive to allow yaw of top mono crystal.
- Two motors to control an "up-down" centre-opening slit.
- Two motors to control an "in-out" centre-opening slit.
- Two motors to control a post-mono centre-opening slit.
- A possible motorised backstop.
- Position monitoring via encoders of all axes and servo control of high resolution drives.
- X-ray count monitors to accumulate counts in several scaler channels for fixed time periods.
- Possible addition of non-standard instruments e.g. Multi-channel analyser (MCA).

4.2 Hutch Area

4.2.1 Pre-Diffractometer

- Two motors to control an "up-down" centre-opening slit.
- Two motors to control an "in-out" centre-opening slit.

- High resolution drive for 2nd stage 2 or 4 bounce mono.
- Two motors to control an "up-down" beam profile slit.
- Two motors to control an "in-out" beam profile slit.
- Position monitoring via encoders of all axes and servo control of high resolution drives.
- X-ray count monitors to accumulate counts in several scaler channels for fixed time periods.
- Possible addition of non-standard instruments.

4.2.2 Diffractometer

The diffractometer will be a medium or heavy duty high resolution instrument similar in concept to that on station 9.4. The main differences will be that the diffractometer will have 4 circles rather than 5 and the Chi circle will not be restricted. In addition, the size of the 2theta and omega axes would be considerably reduced.

- Diffractometer table X movement on rails.
- Diffractometer table Y movement on rails.
- Diffractometer table Z movement (height).
- Possible high resolution alpha drive for magnetic scattering.
- High resolution twotheta circle drive.
- High resolution omega circle drive.
- High resolution drive for chi circle.
- Drive for phi circle (specimen).
- Positioning drive on chi in X axis.
- Positioning drive on chi in Y axis.
- Positioning drive on chi in Z axis.
- High resolution Rotary table analyser drive (triple axis).
- Low resolution drive for the detector system e.g. scintillation counter.
- Position monitoring via encoders of all axes and servo control of high resolution drives.

- X-ray count monitors to accumulate counts in several scaler channels for fixed time periods.
- Possible addition of non-standard instruments e.g. Cryostat.

5 Hardware Requirements

5.1 Motor Drivers

For those axes demanding high resolution, they must be capable of being driven to an accuracy of 0.1 millidegrees. To achieve this, the high resolution encoders must be directly attached to the driven axis rather than the motor shaft. This then requires a system capable of servo-driving and encoding 3,600,000 pulses per degree. This is the maximum resolution used to date on SRS stations and has been very successful on sound mechanical systems. The other axes will be driven adequately using a more conventional stepper system. All axes require an incremental encoder to provide position feedback at least, but also position correction and servo control (closed-loop) or pseudo closed-loop where possible. Lower resolution motors are likely to use encoders on the motor shaft at resolutions typical of a few thousand pulses per revolution.

5.2 Scalers

A sufficient number of scaling channels must be present to easily allow for the initial number of detectors planned. There must also be enough for later expansion and increasing the number should be a routine task not involving major modifications to the control software i.e. no recompilation should be needed. An upper limit of 32 channels is believed to be sufficient.

Each channel must have a sufficient count capacity for the time periods which are going to be used. A 24-bit capacity is believed to be sufficient allowing 16,777,215 counts per timing period per channel. Timing periods would be well within the range of 1 millisecond to 100,000 seconds.

5.3 Non-Standard Instruments

This cannot actually be specified at present but the system must be built in such a way as to be modular and easily expanded to include temperature controllers and MCA's connected via RS232 and additional bus-based instrumentation with minimum software effort.

6 Planned Hardware

It has been decided to use the VMEbus as a platform for data acquisition computing and electronics. This replaces the existing CAMAC, but where necessary CAMAC can be driven from VME to use e.g. specialist CAMAC modules. Within VME, cards must be provided for counter-timing, stepper motor driving (~ 2000 steps/rev) and probably DC-servo motor driving (3,600,000 steps/rev or resolution of 0.0001 degrees). Experience has shown e.g. on the 1.1 Mono, that steppers are unreliable at the high resolutions required by diffractometer axes and can never be truly "closed-loop".

6.1 Counter-Timers

It is proposed that the EC738 and EC740 VME cards are used which will be produced and fully supported by DL staff. They will certainly be in use elsewhere on the 2nd Wiggler. See document "Status of the DL VME Scaling Cards EC738/EC740"[5] for functions and limitations, together with the full technical manuals from I.Sumner[2,3]. Other alternatives exist but they are generally complex to program or restricted in function.

6.2 Stepper Motor Controllers

There are two main VME choices, the Daresbury stepper (8 encoded axes) and the Oregon VME4-4 (4 encoded axes). See document "Evaluation of Oregon Systems VME Stepper Driver"[4]. The DL card has had a lot of problems but will probably be used elsewhere on the 2nd Wiggler. The Oregon card has no software support but does offer some extra features over the DL card e.g. microstepping, position maintenance and user-units but these probably would not be needed. If the DL card proves to be fully usable then this is believed to be the best option.

6.3 DC-Servo Motor Controllers

Investigations are underway to find a replacement in VME for the McLennans PM300 series which uses RS232. Commercial alternatives do not always provide battery backup, optimisation software, high-resolution encoder input and easy connectivity to servo amplifiers. The best card seen so far is the PEP VIMC which includes a software driver and multi-card syncronisation (2 axes per card) but no battery backup. The PM300 would work from a spare terminal port in the VME computer and so would bypass VME altogether. The evaluation of VME alternatives for dc-servo control is underway and near to completion.

Part II

SOFTWARE

1 Principles

- The program must be as flexible and extensible as possible to support the diverse science that is planned.
- The program will be based around a powerful command line interpreter (CLI) giving access both high and low level functions.
- Access to the CLI can be direct from keyboard or from command files. Processing of commands by the CLI then results in access to lower layers of the program which may be hardware specific.
- The hardware i/o and angle calculation functions are available as basic building blocks (low level functions) which can be built, as required, into macros e.g. new scans with no need for software recoding.
- The most demanding high level functions will be hard-coded to improve performance throughout the life-cycle of the program.
- Hardware independence will be maintained, whenever possible, such that e.g. all motor driving is controlled by a single driver routine. This means that an internal representation of the hardware units will be used by the body of the program. New hardware and new user interfaces may thus be easily added later.
- Extensive use of configuration files will be made to avoid repeated trivial code modifications.
- Software prototypes will be provided. A standalone CLI will be initially made available on the Convex C220.
- All code will have inbuilt diagnostic and debugging modes for testing and debugging. The prototypes will initially run exclusively in these modes.
- All program design and documentation will be carried out using a computer aided software engineering tool (CASE) under Unix.

2 Program Data Structures

The software must have provision to store a number of data structures derived from known sample parameters and experimentally-derived parameters. These are required to be memory resident. It must be possible to load, edit, display and transform these values as simply as possible and in a spreadsheet-like fashion, where possible. Space for these items should be dynamically allocated at run-time.

2.1 Hardware Related

- Base VMEbus addresses of cards involved.
- Type, name and number of detector channels active in experiment. Provision for locking and password-protection of motors. As well as scalers, there may be area detectors such as a wire chamber or an imaging plate.
- Name, number and type of motor axes active in experiment, together with limits, offsets and units.
- Device dependant motor parameters e.g. hardware register offsets.
- Current motor positions.

2.2 Experiment Related

- Sample UB orientation matrix (including optional U input by angle).
- Sample crystal structure information.
- Sample environment information e.g. wavelength, temperature, pressure etc.
- Experimental Mode information to comprise geometry e.g. bisecting mode.
- Reflection-list to comprise records containing HKL Miller indices, diffractometer angles, intensity (at point or of peak) and time interval. It should be possible to manipulate it as a simple spreadsheet, and also use as input to a graphical display option.
- Reflection-scan data will be stored which provide input to the reflection-list records i.e. integrated peak intensities. This (DATCOL) data will be normalised and in fixed format so will differ from the usual scan-data which will be found in an SRS file on disk.

- The last record of the reflection-list will be available for fast recall.
- Scan parameters to define variables which will be changed e.g. hkl, and dependant data e.g. intensity. This will also specify the items to be displayed during data collection.
- Where possible, scan data itself will be memory resident. This particularly relates to short setup scans.

3 Program Files

There will be two classes of program files. Firstly, some will contain configuration information which can only be modified by authorized Daresbury staff (as defined by file ownership). Secondly, some files will be specific to each user e.g. command files, and these will be owned by the user (in his directory) and so their maintenance is the responsibility of the user.

In general all of the data structures mentioned above will be derived or have the ability to be derived from disk files, and be written to them. In addition, a few others will exist, as follows.

- Master configuration file, in current directory, to provide general environment-specific information e.g. terminal type, and full file names of all other files used by the program (hardware and experimental).
- Raw command files may be input to the CLI using redirection to carry out repeated operations, setup macros etc. (see below).
- Preprocessed command files may also be input. These will have been fully expanded with limit checking by the main body of the program, including validation of all points specified in scans. This applies both to normal scan and reflection-scan descriptor files.
- Scan-data ascii file to contain standard DL namelist format "SRS header" followed by program specific header of all relevant parameters (as defined by scan parameter data structure). The data itself will be in column format.
- Crystal structure file to contain LAT, UB, WL, extinc and spacegroup.
- Central Help file to contain enough information to allow the program to be run when other support may not be available. This will contain information from both Daresbury computing and science staff.
- A brief Help file may also exist to give a quick guide to the syntax of all commands available from the CLI.

- A log file to record important events and messages. This will include comments which the user may add.

4 Command Line Interpreter

The actual interpreter command set is outside the scope of this document and will naturally expand and evolve with life of the program. The language can be run directly from the keyboard, or from command files, allowing both interactive and programmed methods of operation. The language is effectively "extensible" through the use of command definitions, and this document details the *lowest* level of functionality provided, with which the standard routines will be written.

All commands can be used either directly, as a single-line input, directly, as part of multiple-line input, indirectly as a line in a macro, or within a program file which is executed. The macro is characterised by being invoked as a single instruction name but is actually a list of commands and logic structures.

4.1 Command Language Notation

Command	: Instruction *Note: case independent*
< Arg >	: Mandatory Argument
< A | B >	: Choice of Argument (A or B)
< A & B >	: Choice of Argument (A and/or B)
(-)	: Optional Format

Example of how a command might look

MOVE < Axis > < AbsAngle | + | - RelAngle > could be used as

MOVE TT 5	: Moves 2Theta axis to 5 position
MOVE TT +5	: Moves 2Theta axis 5 from current position positive
MOVE OM -2.5	: Moves Omega axis 2.5 from current position negative

4.2 Variable Types Supported by Program and their Notations

- TotalCounts : Integer
- .Peak : Floating Point
- Points[3] : Integer Array
- #Vector2 : Vector Array
- %Matrix : Matrix (3x3)
- $Line : Character String
- !Finished : Boolean, including error status

Variables can either be global to the entire program, ie those provided as part of the data acquisition package, or local with "subroutines" defined by a user or programmer. Global variables would include such things as the date and time strings, and degree and HKL positions of each motor.

4.3 Logic Constructs

Program flow control commands.

IF condition *THEN* < *COMMAND* > *ELSE* < *COMMAND* > *ENDIF*
: if... then... else

REPEAT < Times > < *COMMAND* > *ENDREPEAT*
: does it n times

WHILE condition *DO* < *COMMAND* > *ENDWHILE*
: condition then loop

DO < *COMMAND* > *WHILE* condition *ENDWHILE*
: loop then condition

4.4 I/O

Reading and writing to files and screen.

OUTPUT (FileName) < Text & Variable >
: write onscreen or to file

INPUT (FileName) < Text & Variable >
: read from file or keyboard

4.5 Conditions

Conditions compare two variables in some way, giving a true or false result depending on the results of the comparison. This would, for example, allow

the start of a scan when a cryostat registered a particular temperature, or terminate a scan should the detector counts suddenly become zero.

The format of conditionals is:

(· Variable1 · *CONDITION* Variable2 ·)
: where *CONDITION* is one of the following

- *EQ* : is equal to
- *NEQ* : is not equal to
- *GT* : is greater than
- *NGT* : is greater than
- *LT* : is less than
- *NLT* : is not less than

Additionally, a complex condition can be built up from simple conditions, in the format

(· Condition1 · *LOGICAL* · Condition2 ·)

using the logical conditions

- *AND* : if both are true
- *NAND* : if neither are true
- *OR* : if either is true

4.6 External File Control

Usage and definition of external command files.

USE · CommandFile · : interpret file as commands

DEFINE CommandName · *GETS* · Variables · *DOES* *COMMANDS* · *ENDDEFINE*
: defines new command

MIMIC · ProgramFile · : creates program file that does exactly what user types in interactively

ENDMIMIC : ends a MIMIC session

SETSECURE : sets secure mode... all moves validated before scan

SETDEBUG : allows tracking of internal variables etc for debugging

4.7 Access to Operating System

Provision must be made for shell commands to be executed and probably also an escape to shell to avoid having to quit and restart the program to obtain a directory listing, for example. This will give access to a system screen editor for some parameter file modifications as an alternative to the general-purpose editor available from the CLI.

4.8 Low Level Functions

For conceptual purposes, program functions are divided into low-level and high-level functions. The former are the lowest-level commands provided which form the basis of higher level functions throughout the program and would not be invoked directly by most users. They may be combined together to form new scans etc. which will not be available elsewhere. This allows the software to act as a simple interpreted experiment-programming language.

4.8.1 Hardware

Commands for controlling hardware, eg motors, counter-timers, or cryostats.

MOVE < Axis · < AbsAngle | (+ | -)RelAngle >
: Move a motor, in degrees

MOVEHKL · AbsHKL | (+ | -)RelHKL >
: Move a motor, in HKL format

COUNT (< Channel >) < Time >
: Detector count for spec'd time

TEMP < Temperature >
: Alter temperature on cryostat or heater

4.8.2 Calculative

Arithmetical operations.

ADD · A > · B > = · C >
: add two values, result held in C

MATRIXADD · #A · #B > = < #C ·
: add two matrices (3x3)

MATRIXTRAN · #A · = · #C >
: transpose a matrix (3x3)

MATRIXINV · #A · = · #C ·

: invert a matrix (3x3)

4.8.3 Manipulative

As well as operating on a matrix as a single entity, it must be possible to access elements of the matrix as scaler data items.

4.8.4 Conversion Routines

Conversion between different representations.

AH · Angle · · HKL angle ·
: Converts degrees angle to HKL

CA · Cartesian · · Angle ·
: Converts cartesian coordinates to degrees

4.9 High Level Functions

These are hard-coded routines written using the minimum command set above, to extend the functionality of the language somewhat, providing standard complex functions. A more complete description of the known options will be provided in a later section.

4.9.1 Scans

These will be hard-coded procedures involving mainly hardware-related operations of angle calculation, motor driving and scaling. A scan-descriptor command file will normally specify the details of operations in the scan. A fixed format of scan (DATCOL) is required for manipulations involving the reflection-list, which is the reflection-scan. As a general rule, however, data will be saved to an SRS dataset and stored in memory where possible. A display will provide graphical and textual information on scan parameters and the acquired data.

4.9.2 UB Matrix Calculations

Calculates UB Matrices from found reflections

UBCALC2 · Reflection1 · · Reflection2 ·
: Calculate UB Matrix from 2 reflections

UBCALC3 · Reflection1 · Reflection3 · · Reflection4 ·
: Calculate UB Matrix from 3 reflections

UBCALC · 20 ·
: Calculate UB Matrix from first 20 reflections

4.9.3 Data Input

Read complex data formats from user directly.

READLAT · Lattice Values · · (FileName)
: Read lattice values from user or file

READUB · UB Matrix · · (FileName)
: Read UB Matrix from user or file

5 High level Function Overview

This is not a complete list but summarises the current state of thought and will be extended as necessary.

- Various hard-coded scans as specified by scan data structure. These will include 1-axis, 2-axis, H, K, L, HKL along a line and a Cone scan.
- Ability to display, load, screen-edit and save all user-definable data structures and control modes.
- A subprogram calculation/HB program to interconvert angles, HKL's and cartesian coordinates.
- Miscellaneous other calculations. Top reflection, Hamilton's eight equivalent positions on the reflection-list. Fixed phi etc.
- Calculations which have an effect on the whole reflection-list. Index, Intindex and Bisecting modes etc.
- UB calculation from 2 or more indexed reflections on the list and the known lattice. U*B possibility and optional U input by angles.

- UB short calculation from all reflections on the list, where no indices are known (left handed). The three shortest and most perpendicular are selected from the list.
- UB least squares. Reflection intindex is needed beforehand.
- Reduced cell Niggli transformation of direct cell parameters, UB matrix and optionally HKL indices on the reflection-list (with "integer" HKL).
- Blindsearch for reflections with results to be written to a file. This will work in association with an Analyze and a Centre function (both also available directly as low-level functions).
- Bragg HKL function to measure single peak and background by HKL or by angle.
- SETHKL to set up an hkl file within hkl limits, taking into account extinctions and motor limits.
- DATCOL to measure a certain number of reflections using Centre and/or Bragg functions. This will use a file written by SETHKL or other means. The profile may be optionally output.

6 On-line Help

- A central help option will be provided with extensive information on operation of both the software and the experiment itself.
- A brief help will be available for all commands generated by typing a command name followed by · CR ·. This will list any qualifiers available.

7 Angle Calculations

Software for these will be provided by G.Oszlanyi based on the methods of Busing and Levy[1]. Close cooperation will be needed with the authors to allow seamless integration of the code from the various programmers.

8 Naming convention

Throughout the program, standard names will be used to define both the motor axes and other parameters involved in scans and sample environment. These will be as self-explanatory and unambiguous as possible and will be

accessed from various program files but should not be changeable by normal station users. There will be both long and short forms of the names available.

9 User Aborts

There must be a mechanism for both a soft and a hard abort. The soft abort will halt the current function, including a scan, and return to the level of the CLI if possible. After a number of commands have been specified, the halted function may be resumed or terminated with tidy-up. The hard abort will stop the executing function and tidy-up as much as possible e.g. close open files, saving data collected already.

10 Error Handling

There will be a mechanism to control the response to errors encountered during execution of a function. Normally the program will pause on error and allow a decision to be made about continuing (as with a soft abort) by the user. In batch operation, it will be possible to force continuation without pausing if feasible. In the case of hard errors, no recovery is possible and the program must tidy-up and halt the function in a clean way.

11 Software Implementation

In order to have a real-time operating system with easy access to the VME hardware together with a high level of support for data acquisition cards, it has been decided to use Microware's OS-9 on the DL standard Motorola 680X0 processor family. This decision requires the use of the C language for the software which is now the chosen data acquisition group Standard for new projects. OS-9 is not a good general purpose operating system as is shown by poor support for languages other than C, a small number of development tools compared to Unix and the lack of demand paging. Ideally, a Unix workstation will be purchased to cater for all data analysis requirements. The two systems will initially be loosely coupled via the site Ethernet but future more close integration is possible.

12 Acknowledgements

Thanks are due to R.Cernik and G.Oszlanyi for providing much of the raw information worked into this document, and for checking its content as written

by the computing staff. Thanks are also due to C.Ramsdale and B.Bowler for involvement in a specification peer review.

13 References

1. Busing,W.R. and Levy,H.A. (1967) Acta Cryst. *22*, 457-464. "Angle Calculations for 3- and 4- Circle X-ray and Neutron Diffractometer".

2. Instrumentation Development (1991) SBT/Instrumentation/M25 Revision 1. "EC738, 32 Channel Multi-channel Scaler".

3. Instrumentation Development (1991) SBT/Instrumentation/M26 Revision 1. "EC740 Time Frame Generator VME Module".

4. Miller,M.C. (1990) Internal report. "Evaluation of Oregon MicroSystems VME Stepper Driver".

5. Miller,M.C. (1991) Internal report. "Status of DL VME Scaling cards EC738/EC740".

6. Vlieg,E., Van Der Veen,J.F., MacDonald,J.E. and Miller,M. (1987) J. Appl. Cryst. *20* 330-337. "Angle Calculations for a five-circle Diffractometer used for Surface X-ray Diffraction."

Part III
APPENDIX

1 Introduction to Single-Crystal Diffraction

If a three-dimensional *single-crystal* is exposed to an incident beam of x-rays, some of the photons are *scattered or diffracted* by successive parallel planes of atoms with no resulting change in photon energy (i.e. *elastically scattered*). This results in a number of diffracted beams exiting the crystal at various scattering angles, all with greatly reduced intensities compared with the incident beam. For each atomic plane, the scattering conforms to *Bragg's Law* such that

$$\lambda = 2d\sin\theta$$

where λ is the incident x-ray wavelength, d is the spacing between the particular parallel atomic planes in question and 2θ (2Theta) is the angle through which the incident beam is deviated.

In a crystal, the basic repeat diffracting unit is known as the *unit cell* and within this a number of *scattering planes* can be constructed. Unit cells with a high level of symmetry contain few such planes, e.g. cubic materials. Only the planes with larger *d-spacings* are usually measured as it is possible to obtain ever more oblique planes which approach being parallel. It is the *diffraction peaks* or *reflections* from these distinct planes which characterise the constituent material by a combination of their positions and intensities. This diffraction pattern can be used to derive the structure of the scattering material or follow changes in the structure in response to changes in the sample environment.

The basic instrument for carrying out single-crystal diffraction experiments is the *diffractometer*, consisting of motorised axes or *circles* which are usually free to rotate up to 360 deg. There are usually *four circles* provided. The sample is mounted at the centre of the diffractometer and can be oriented by the *Omega*, *Chi* and *Phi* circles such that a photon detector on the *2Theta* circle (sometimes called Delta) can measure the diffracted photon counts for particular reflections. For some applications in *magnetic diffraction*, an extra circle is needed (*Alpha*) to rotate the detector in a plane perpendicular to the 2Theta axis. For highest resolutions, a motorised analyser crystal is mounted on the 2Theta arm, giving six circles in total. The combination of monochrometer, specimen and analyser is called *triple-axis mode*.

Movements of the diffractometer are usually considered to take place in units of diffracted-beam space rather than units of diffractometer circles, and this

is known as *reciprocal space*. The *Miller indices* h,k,l define the reciprocal lattice points for a crystal. The calculations between real-reciprocal space depend on the unit cells of the sample being in a standard orientation on the diffractometer. The U and B matrices (both $3 \cdot 3$) provide a mapping from the actual positioning on the diffractometer to correct alignment in reciprocal space to allow the standard calculations to be used for positioning. The B matrix transforms a cell with non-90 deg angles (e.g. monoclinic) to orthogonal axes, while the U matrix orients the resulting orthogonal axes to a standard frame.

Usually something is known of the sample structure or at least its chemical composition and this greatly speeds the following investigations. Much of the early experimental work with a sample is concerned purely with deriving a *UB matrix* (the product of U and B). Once this is known, then particular reflections can be scanned and the data used to further refine the UB matrix or to examine the fine structure of the material in detail.