

**Development of Data Acquisition and Analysis Software
for Multichannel Detectors**

Y. Chung

Princeton Plasma Physics Laboratory

P.O.Box 451

Princeton, NJ 08543

ABSTRACT

This report describes the development of data acquisition and analysis software for Apple Macintosh™ computers, capable of controlling two multichannel detectors. With the help of outstanding graphics capabilities, easy-to-use user interface, and several other built-in convenience features, this application has enhanced the productivity and the efficiency of data analysis.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

1. Purpose

This application was developed to acquire, analyze, and archive spectroscopic data from up to two multichannel detectors, and was used in combination with the VUV-visible duo multichannel detectors in an X-ray laser experiment. The development of the VUV-visible duo multichannel detectors is described in a separate report and the experimental result obtained with this application, measurement of the quenching of Einstein A coefficients, is published elsewhere.¹ Neither the data acquisition system nor the experimental result are presented in detail in this report. Since this report is not intended to be a user's manual, only general framework and programming principles are presented, including a somewhat detailed description of techniques rather than specifics of how to use the application. To illustrate the techniques and the programming considerations, C language, which is the language in which the application is written, is used throughout. Graphics are used wherever possible.

This report consists of 1) Purpose, 2) General Discussion, 3) Text Editing, 4) Data Acquisition Procedure, 5) Data Display, 6) Data Analysis, 7) File Management and Manipulation, 8) Miscellaneous Features, and 9) Summary.

2. General Discussion

In recent years, micro and personal computers have gained importance in scientific applications and are slowly replacing mini and mainframe computers in small-size experiments. Due to technical breakthroughs over the last decade, micro computers with appropriate hardware/software combination are now capable of sophisticated tasks. However, the use of micro computers for experimental projects has lagged mainly because of need for high speed, extensive storage devices, and multitasking capabilities. So far,

even the most advanced micro computers cannot, even closely, satisfy all these needs. However, in some mid and small-size experiments, these micro computers are quite suitable to carry out data acquisition, analysis, and instrument control.

The application described in this report was designed for the Apple Macintosh™. With the aid of over 500 functions built into the ROM, software development can be greatly facilitated and enhanced by the Macintosh's consistent user-friendly interface. This application is no exception. A great effort was made to strictly follow the user-interface guidelines² by using windows, palettes, dialogs, etc. The user-interface of this application is illustrated in Fig. 1. Multiple windows are allowed and limited only by the amount of available memory. The window's size and viewing area can be easily adjusted with the growicon and the scroll controls, respectively.

The Macintosh provides for extensive user control. This is made possible through several so-called 'Managers' and their supporting functions in the ROM (Read Only Memory). The Event Manager checks all the user-driven events, such as mousedown events or keydown events, and posts them in the event queue, which is accessible by the application. Then, while going through the main event loop, the application checks the availability of the events that should be handled by the application, retrieves the appropriate event, and executes the commands, if any, according to the instructions. The main event loop looks like :

```
do {
    if (GetNextEvent (everyEvent, &myEvent)) {
        switch (myEvent.what) {
            case mouseDown :
                domousedown();
                break;
            case activateEvt :
```

```

        doactivate();
        break;
    case updateEvt :
        douupdate();
        break;
        .
        .
        .
    default :
        break;
    }
}
} while (!finished);
ExitToShell();

```

As shown above, the `GetNextEvent()` function determines whether an event is available. If available, it removes the event from the event queue and stores it in the variable 'myEvent'. The application inspects the content of this event record and branches into one of several different functions accordingly. This process goes on indefinitely until the user gives the 'Quit' command. Then the variable 'finished' is set to TRUE and the application quits the loop and usually exits to the 'Finder' by executing `ExitToShell()`. Thus, the Event Manager function, `GetNextEvent()`, plays a central role between user action and application response, whereby the user maintains the control of the computer.

3. Text Editing

If the computer does a certain operation and keeps the result to itself, it would be useless to the user. It must be displayed on the screen or output to some other device. Moreover, as emphasized in section 2, the user must maintain as much control as possible over the computer operations, which is particularly important when an operation takes a long time and the user wishes to abort it. This may be done either by tracking user actions

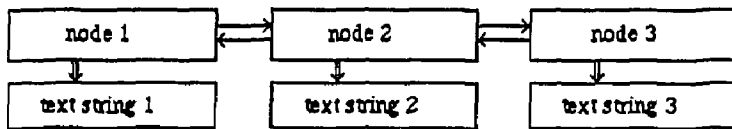
all the time during the operation or, in a more natural way that follows the user interface guidelines, by doing the operation incrementally during each pass through the main event loop. In some cases, the combination of these two may be appropriate. The following is a description of how the application handles text strings produced either as a result of analysis operations or simple text editing.

The application supports all basic text editing operations: keyboard input, cutting, copying, pasting of texts, and undoing of the last text editing operation. Each text string can be displayed using a different font, size, or style, as selected from the menu. It can also drag the displayed text to any place inside the window as described in section 6. A window usually has more than one text string. The user creates a text string simply by clicking the mouse at the position where he wants to write the text. This necessitates the record keeping of all the text strings which belong to a window, so that the application can keep track of them. This application employs the technique of 'linked text strings' to facilitate using the TextEdit Manager residing in the ROM, which is capable of most of the basic level text editing. Whenever a new window is opened, the application creates a chain consisting of two text strings, i.e., text nodes. The structure definition for this node is:

```
typedef struct TENODE{
    TENODE *lastrecord;
    TENODE *nextrecord;
    TEHandle text;
    .
    .
    .
} tenode;
```

The structure members 'lastrecord' and 'nextrecord' hold the addresses of the last and the next text node, respectively. 'Text' is a handle (that is, pointer to a pointer) to

TERecords, which is a structure defined by TextEdit Manager. Any text editing operation is handed over to this TextEdit Manager together with the address of the text node, or equivalently, a handle to the text. To illustrate this graphically, the text nodes are linked to each other like the following:



The TERecords hold the left, top, right, and bottom coordinates of the rectangle inside which the text is displayed. Thus, when the user clicks the mouse inside the window, the application searches through this link starting at the first node for the text on which the mouse was pressed. If a text node is found, it becomes the current text and a caret blinks at the insertion point. If not, the application creates another text node, makes it the current text, and attaches it to the link. The caret will blink at the beginning of the text. Afterwards, all text editing operations will be directed to this current text until the user clicks the mouse outside the viewing rectangle of this text. The application then repeats the procedure.

Once a text node is made current, the user can directly edit the text using the keyboard or the mouse. The data acquisition and analysis application is designed to use this text editing capability to display the results of a certain operation, such as integration and calculation of a wavelength. These text strings are saved to the file with all the information needed to redisplay the text exactly as it was, including the font, size, style, and the coordinates of the pixel and intensity corresponding to the location of the text. The

pixel and intensity coordinates are needed to relocate the text string when the user either expands or shrinks the range of the pixels.

4. Data Acquisition Procedure

This experiment began about two years ago with a multichannel detector on a VUV spectrometer and a photomultiplier tube on the visible spectrometer, each viewing the CO₂ laser- produced plasma from opposite directions. The data from the VUV multichannel detector were obtained and analyzed by this application and the visible line intensity was recorded on an oscilloscope. The visible spectrometer was upgraded to a multichannel spectrometer about a year later for greater accuracy in measuring the quenching of spontaneous emission coefficients. At present, the data acquisition system consists of a VUV and a visible multichannel detectors whose data can be collected either simultaneously or separately as controlled by the software. The schematics of this system are shown in Fig. 2.

Since the Macintosh Plus has only two serial communication ports, a switching box was connected to the modem port to allow for alternating between the VUV and the visible detector controllers for data collection. This was possible due to the 512 kbytes of memory in these detector controllers, where the incoming spectrum was first stored and retrieved later. When only one of the spectrometers was in operation, this scheme was not used: The spectrum was sent directly to the memory in the computer.

Recently, a hardware addition which has IBM-PC/AT style expansion slots for Macintosh Plus has become commercially available with a set of software to be used in combination. This uses the SCSI (Small Computer System Interface) parallel communication port in the Macintosh Plus and could be used in place of the switching box

described above. Even though this was not used in the experiment described in Ref. 1, the application was later updated to drive this hardware, which is being used in another X-ray laser experiment.

To synchronize the detector controllers and the laser pulse, which produces a plasma on the target surface, an optical pulse with a delay of 1 msec is provided by the switching box to the laser triggering circuitry. This delay time was optimized for minimal noise on the spectra. The application, which was originally developed with only one spectrum in mind, was also updated to handle the two spectra simultaneously. Figure 3 shows the dialog which checks for errors during the entire course of data acquisition. This dialog simultaneously monitors such user actions as mouse click or key strokes inside the dialog box and communicates with the detector controllers. In case of an error, it stops the procedure, aborts the detector controllers and waits for further instruction from the user. Of course, the user can abort data acquisition either with a mouse click or with an escape key at any time. If there were no errors or no user-initiated abort, the data acquisition will terminate successfully, and the spectra will be stored in the computer memory and displayed on the window. Figure 4 shows one set of VUV and visible spectra thus obtained. The upper half of the window shows the VUV spectrum and the lower half the visible spectrum. Either of the spectra can be drawn on the full window with a click of the mouse on one of the control buttons located in the upper right corner or with an equivalent combination of the key strokes from the keyboard.

5. Data Display

With the successful termination of the data acquisition procedure, the data are immediately displayed with the full range of pixels and with the maximum intensity

automatically adjusted. Usually, the redrawing of the entire window content is done indirectly by forcing the system to post an update event. The system posts an update event for a window whenever its content needs redrawing, and the application informs the system that a window needs refreshing by declaring its content to be invalid. The application then retrieves the update events during a subsequent pass through the main event loop and redraws whatever windows the update events are posted for. During the updating, only the areas declared invalid, implicitly or explicitly, are redrawn. Even though the application does not explicitly declare a particular area invalid, if it needs be redrawn because, e.g., it had been covered by another window, the system declares it invalid and adds it to the update region. However, the system simply tells the application which window needs updating, and the rest of the job, actual drawing, is the responsibility of the application. In this application, the so-called 'bit copying' is extensively used when the content of the window needs redrawing without any change, e.g., after closing a dialog or when scrolling. Since there are a lot of dialogs used throughout the application, this method is very useful, especially when there are several windows present on the screen. Once the spectra are displayed inside the window, the user can select or change the scope of display by a variety of ways as described below.

The user can choose the scope of display by first pressing the mouse at the top left corner of the rectangle whose interior is to be expanded, and then dragging the mouse to the right end of the pixel range while keeping it down. Then a thick-lined rectangle is drawn around the area to be expanded, asking the user to confirm or cancel the choice. If the user clicks the mouse again inside the rectangle, its interior will be expanded and redisplayed, while if the user clicks the mouse anywhere outside it, the rectangle disappears and nothing happens. Figure 5 shows the thick-lined rectangle that the user has chosen for

expanded display. The choice of the display rectangle is done inside one pass through the main event loop. After the rectangle is chosen, the application does not halt but goes through the main event loop many times, waiting for the mousedown event. However, the user is still allowed to do other things as well, such as opening a new window or a dialog. The menu items corresponding to actions that are not allowed are dimmed, thus prohibiting the user from doing so.

Another way to change the scope of display is to use the horizontal bars below the spectra and the vertical black bars left to the spectra. The horizontal bars are the pixel range bars and the vertical bars are the intensity range bars. The user can control the ranges by pressing down the mouse, dragging it while keeping it down. The cursor is changed to the shape of a finger, a right-or a left-hand finger according to the location of the mouse. The end pixel numbers and the maximum intensity are changed to new values. If the user opts for fixed pixel range or fixed maximum intensity for all windows, it can be done by pressing the mouse on the number to be fixed. The number is surrounded by a solid-lined rectangle, meaning that number will be used for all windows from then on. This can be undone by pressing the mouse on that number again. A dialog is also provided to do all this. This dialog can set the end pixel numbers, the maximum intensity and the shot number, and tell the application whether to display the background spectrum or line spectrum.

In case there are several windows open simultaneously, the size of the window can be changed by dragging the growicon located at the right bottom corner of the window. If the window becomes smaller than its maximum possible size, the horizontal scrollbar and/or the vertical scrollbar are made active, thus allowing the user to scroll the window content up and down, to the left and to the right. The zoombox at the right top corner

enables the user to switch between the reduced size and the full size window.

6. Data Analysis

Data analysis is an extensive part of this application. In fact, the data acquisition procedure described in the previous section is only reading the spectra from the detectors and storing them in the computer memory. Once the data are transported to the memory, the application can perform all the operations the user wants and display the result on the window. These operations include: simultaneous display of pixel, intensity, and wavelength following the motion of the mouse, writing them at the position wanted, data smoothing, and integrating the line intensity over the spectrum profile.

The top right corner of the rectangle in which the spectrum is displayed is reserved for the display of the pixel number, intensity, and wavelength corresponding to the current mouse position. When there are no more events available, the application follows the motion of the mouse. If the mouse is inside the spectrum rectangle, it finds the pixel position and intensity, calculates the wavelength, and displays them in the reserved area. For efficiency in analyzing the data, the application provides the user with a choice of 'modes'. Modes are selected from a palette, which is a collection of symbols, usually enclosed in rectangles. A symbol can be an icon, a pattern, a character, or just a drawing, that stands for a particular mode. With palettes, changing from one mode to another can be done almost instantaneously, and the user can always see at a glance which mode is in effect. The user usually chooses a mode by pressing the mouse on one of the symbols in the palette and the application responds by visually darkening the symbol. The symbol remains darkened until another mode is chosen, thus reminding the user of which mode is in effect. This application has five modes, and the palette is located in the top left corner of

the window, as shown in Fig. 1. Each window has its own palette. It should be remembered that regardless of which mode the application is in, it still goes through the main event loop to get the next available event, to which it may, however, respond in different ways in different modes.

The first mode is text editing. The user can create, choose, or edit a text string in this mode, besides other basic operations common to all modes, such as scrolling, changing display range, and selecting background level. All menu items allowed are enabled and those not are disabled. In the second mode, the application automatically writes pixel, intensity, or wavelength, or any combination of these as set by the user, at the position where the mouse is pressed. In the third mode, the cursor changes into the shape of scissors which cuts away sharp spikes or other noise in the spectrum. In the fourth mode, the user can move an existing text string to another position inside the window. The fifth mode is the integration mode, which involves a series of steps to complete one operation. When the user selects this mode, the application first asks where to write the result of the integration and the user responds by clicking the mouse at the position wanted and pressing the tab key or the 'ok' button provided on the window. A new text node is created and is editable and movable as any other text string. Then the cursor changes into the shape of a cross in a circle, with which the user specifies the starting and end points of integration. The result is immediately written at the position previously chosen by the user. During each step, the user can exit the mode by pressing the 'quit' button or another mode symbol.

Another useful tool for data analysis is the dialog for the wavelength calibration of the spectrometers. The calibration parameters are stored in the application as resources and can be modified through the dialogs, which, given the pixel number and the wavelength of

an already known line, recalculates one of a few calibration parameters. The user specifies which of them should be recalculated and the new parameters are used for all calculations of wavelengths thenceforth.

7. File Management and Manipulation

The File Manager residing in ROM provides all the necessary operations needed to handle communication between an application and files on block devices, such as disk drives. The operating system also allows the use of an external file system other than the File Manager. However, the introduction of a new UNIX-style file management system called HFS (Hierarchical File System), which is a major feature of the 128K ROM version of the File Manager, obviates such a need.² The new version significantly speeds up access to large volumes compared to the old 64K ROM version called MFS (Macintosh File System or Flat File System).

This application fully uses the new HFS in various operations on files. The Standard File Package, also residing in ROM, provides a convenient way for the user to select a file that is contained in a folder. The Open Dialog, which is displayed when the user gives the 'Open' command from the menu, lists only the files contained in a single folder. The user can also indicate which folder to choose a file from. Using HFS, data taken on different days can be stored in different folders or directories, automatically dividing a large number of files into several groups. This feature is especially useful when the files are stored on large capacity hard disks. Under MFS which lists in the Open Dialog all the files stored on a volume, the user might have to search through thousands of files. This exhaustive, linear search of all files is extremely time-consuming. To circumvent this difficulty, hard disk manufacturers in the past divided the disk space into several volumes,

whereby the File Manager recognized a single hard disk as a collection of smaller volumes, or equivalently, disks. The 128K ROM version solves this problem by using the notion of 'working directories,' which can be called pseudo-volumes, and allows existing applications to operate with HFS.

Besides the standard file operations, such as reading from and writing to disks, this application lets the user add several shots, subtract a shot from another, and multiply or divide the data with constant numbers. If the line intensity is too weak or the signal-to-noise ratio is too small, several shots can be added to increase the intensity at the same time as reducing the noise. To see the effect of a change in the experimental conditions, the shot before the change can be subtracted from the shot after the change, thus enhancing the differences. The record of all arithmetic operations on the data is automatically recorded and stored into the file, and can be printed or displayed as, e.g., 'Shot xxxx + Shot yyyy' as an editable item in the Experimental Setup Dialog (see Fig. 6).

Other useful features regarding the file management are the en masse listing and updating of certain stored parameters of all shots in a folder. For instance, the user can view the listing of detector positions and target species, and quickly locate the file to open. Or, the calibration parameters of all shots stored in a folder under HFS can be updated altogether, which relieves the user of the tedious task of opening the files individually and changing them one by one.

8. Miscellaneous Features

Multiple windows with adjustable size and position are allowed, whose number is limited only by the size of available memory. A list with the titles of all open windows is provided as menu items from which the user can bring a particular window up to the front.

The first menu item indicates the size of free memory. The second item enables the user to close all open windows and the third item to save all changes made to the windows. The order of the remaining menu items is the same as the order of the windows, that is, from the front to the back. The list is automatically updated whenever there is a change to the number of or the order among the windows, the fifth item always corresponding to the frontmost window, the sixth item to the second one and so on. The fourth item is a dotted line separating the first three items from the remainder.

The Finder interface, a part of the Segment Loader, provides a unique way of linking the shell, usually the Finder itself, and the application by using the resources 'BNDL' and 'FREF'. Several files can be printed in sequence or opened and displayed simultaneously. The only restriction is that these files should be in the same folder. Hundreds of shots (actually an almost unlimited number) can be printed without the user attending the computer. The user can check the number of remaining files to be printed, stop printing temporarily, or abort it altogether by pressing one of the buttons on the dialog. The dialog also shows the name of the file currently being printed. This method of printing is the most commonly used with this application, even though the data can be printed directly from the window. The application also allows the user to continue working while printing, when the computer constantly switches between the printing and the execution of the main event loop until the printing is finished. In this case, the main event loop is the background procedure which is run at every opportunity while the computer is waiting for the printer to finish. This process, however, is not very fast. Both printing and execution are considerably slowed down, and the application responds to mouse clicks and key strokes with a noticeable time lag.

9. Summary

The development of multichannel data acquisition and analysis software has been described. Throughout the development phase, the Macintosh user interface guideline has been followed as closely as possible. As a result, this application provides an interface easy to use and consistently user-friendly. This application had been developed initially for one multichannel detector and the prototype had only a limited capability. About a year later, another multichannel detector was added to the data acquisition system and the application underwent a major enhancement and refinement, including the ability to obtain and analyze data from two multichannel detectors simultaneously. This application is not designed as general purpose data acquisition software. Its capabilities, for now, are quite restricted in the sense that it cannot easily be expanded to accommodate other types of instruments. This can be achieved only through a source code level modification. However, this application provides an extremely useful starting point for future expansion.

Acknowledgment

This work was supported by U.S. Department of Energy Contract No. DE-ACO2-76CHO3073.

References

- ¹ Y. Chung, P. Lemaire, and S. Suckewer, Phys. Rev. Lett. 60, 1122 (1988); Princeton Plasma Physics Laboratory Report No. PPPL-2472.
- ² Inside Macintosh, vols. 1, 2, 3, Apple Computer, (Addison-Wesley, Reading, 1985)
vol. 4, Apple Computer, (Addison-Wesley, Reading, 1986)

Figure Captions

- Fig. 1 The palette from which the user can choose a mode. The symbols represent: text editing mode, data display mode, scissors mode, text moving mode and integration mode, respectively. Each window has its own palette. The windows are also shown with horizontal and vertical control bars to scroll the content of the window. At the bottom right corner is the growicon with which the user can adjust the size of the window and at the top right corner is the zoombox.
- Fig. 2 (a) Schematics of Data Acquisition System for VUV-Visible Duo Multichannel Spectrometer (b) Experimental Setup.
- Fig. 3 The Data Acquisition Dialog. The left control button is used to trigger the laser or to restart an aborted shot, and the right one to abort the shot or to quit the data acquisition procedure.
- Fig. 4 An example of a window displaying both VUV and visible spectra. The upper is the VUV spectrum and the lower is the visible spectrum. Either of them can be displayed in full by pressing one of the control buttons in the top right corner.
- Fig. 5 The thick lined rectangle represents the area chosen by the user for expanded display. If the user clicks the mouse again inside the rectangle, the interior of the rectangle will be expanded and redisplayed, while if the user clicks outside, the rectangle disappears and the operation is canceled.

Fig. 6 The Experimental Setup Dialog. Detector positions, target species, target position, and comments are recorded through this dialog and stored to disk later.

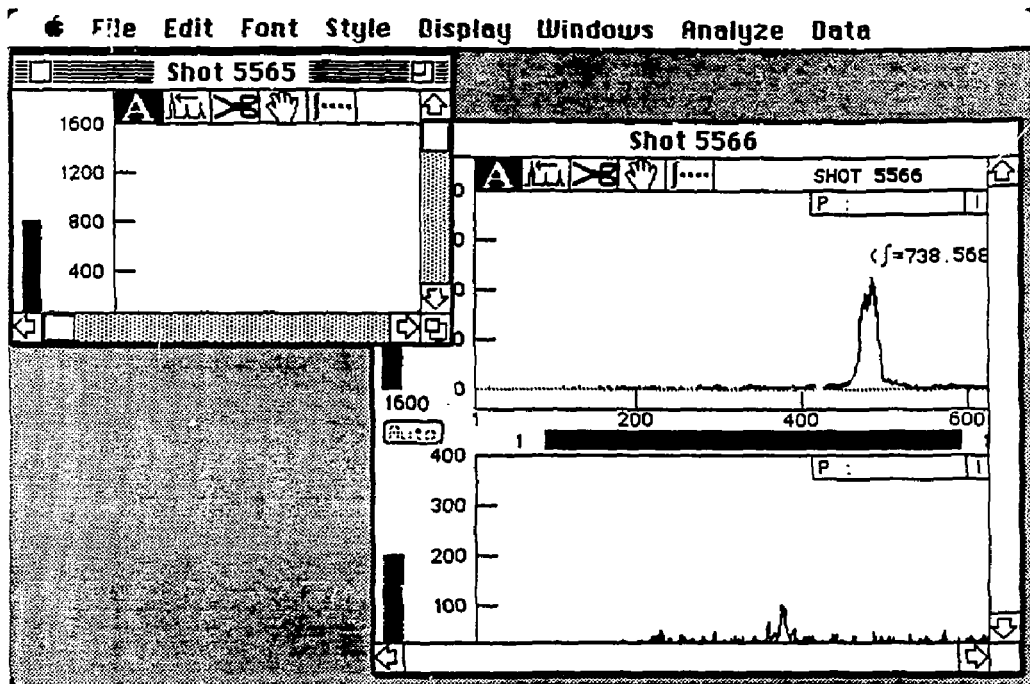


Fig. 1

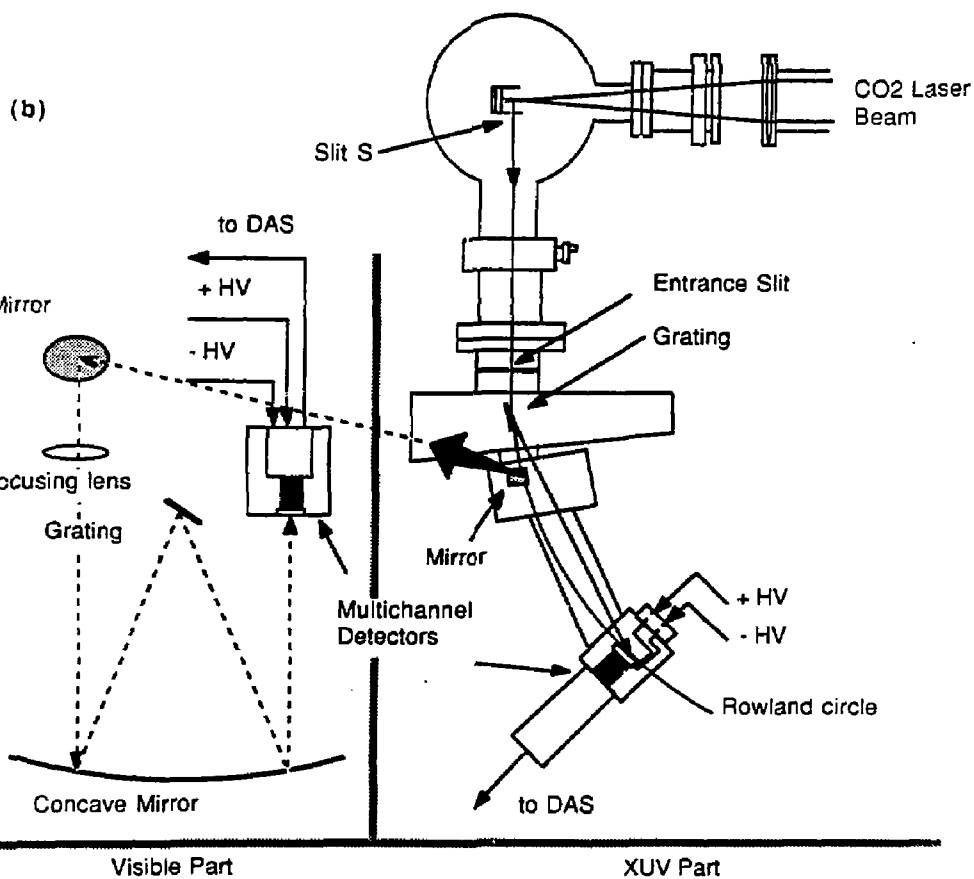
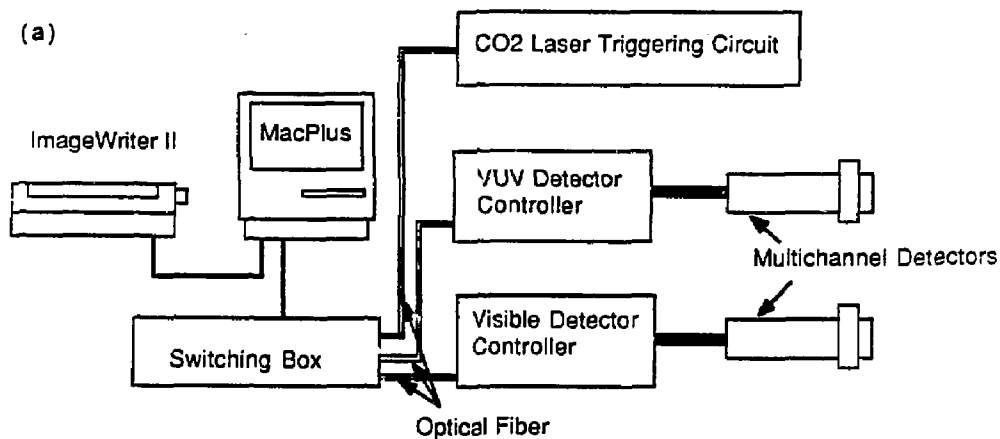


Fig. 2

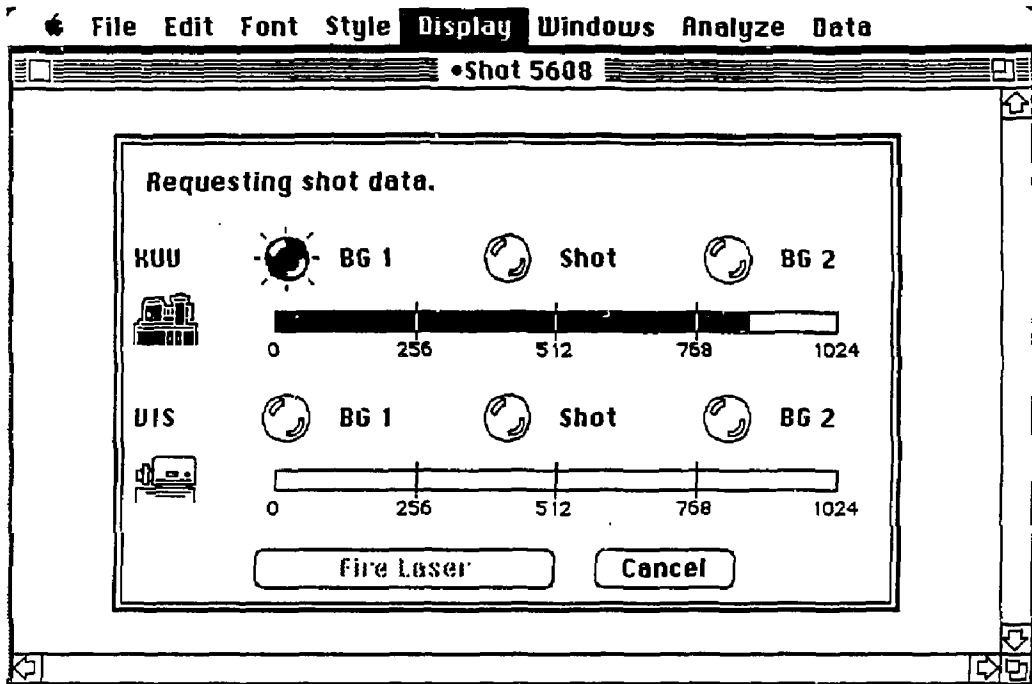


Fig. 3

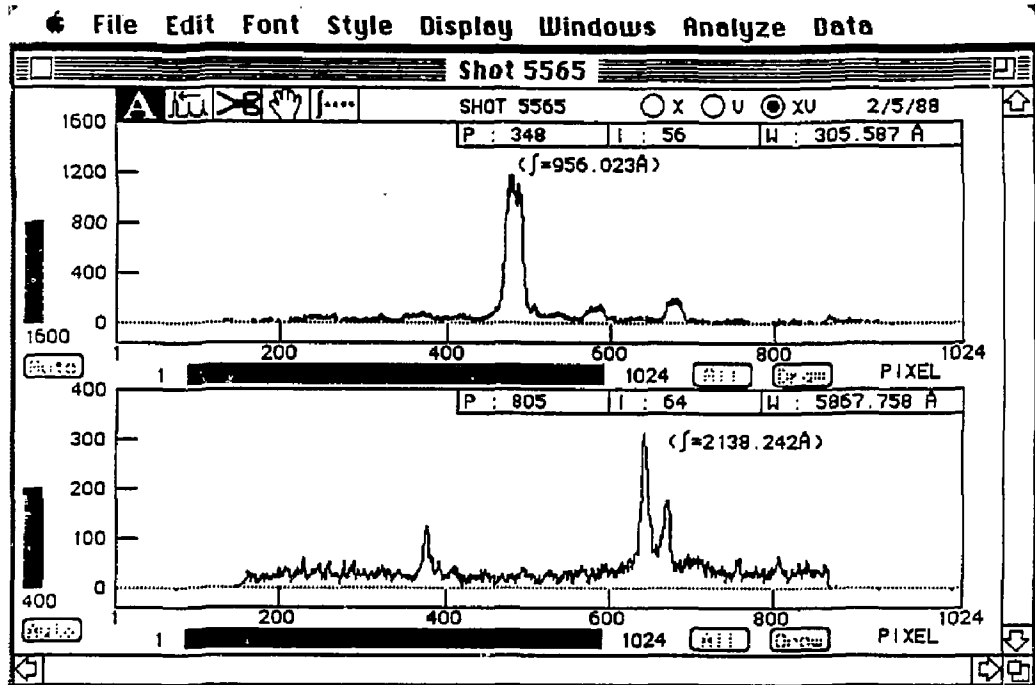


Fig. 4

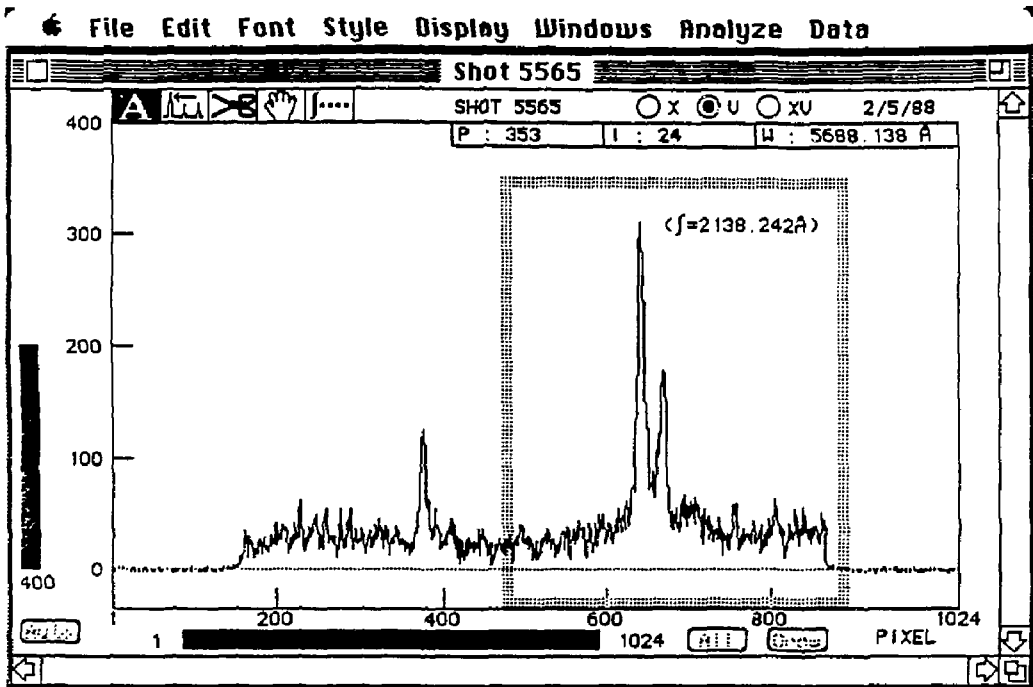


Fig. 5

File Edit Font Style **Display** Windows Analyze Data

Shot 5565


Experimental Setup Data

XUV Spectrum : McPherson 1.5m

☒ Detector Position "
☐ Wavelength Å Halfrange Å

Slit Width μm Grating (/mm)


Translation " Rotation "

ON  OFF

VISIBLE Spectrum : Jarrel-Ash

☒ Detector Position Å
☐ Wavelength Å Halfrange Å

Slit Width μm Grating (/mm)

ON  OFF

Target Comment

Step

Fig. 6

EXTERNAL DISTRIBUTION IN ADDITION TO UC-20

Dr. Frank J. Paoloni, Univ of Wollongong, AUSTRALIA
 Prof. M.H. Brennan, Univ Sydney, AUSTRALIA
 Plasma Research Lab., Australian Nat. Univ., AUSTRALIA
 Prof. I.R. Jones, Flinders Univ., AUSTRALIA
 Prof. F. Cap, Inst Theo Phys, AUSTRIA
 Prof. M. Heindler, Institut fur Theoretische Physik, AUSTRIA
 M. Goossens, Astronomisch Instituut, BELGIUM
 Ecole Royale Militaire, Lab de Phys Plasmas, BELGIUM
 Commission-European, Dg-XII Fusion Prog, BELGIUM
 Prof. R. Boucique, Laboratorium voor Natuurkunde, BELGIUM
 Dr. P.H. Sakanaka, Instituto Fisica, BRAZIL
 Instituto De Pesquisas Espaciais-INPE, BRAZIL
 Documents Office, Atomic Energy of Canada Limited, CANADA
 Dr. M.P. Bachynski, MPB Technologies, Inc., CANADA
 Dr. H.M. Skarsgard, University of Saskatchewan, CANADA
 Dr. H. Barnard, University of British Columbia, CANADA
 Prof. J. Teichmann, Univ. of Montreal, CANADA
 Prof. S.R. Sreenivasan, University of Calgary, CANADA
 Prof. Tudor W. Johnston, INRS-Energie, CANADA
 Dr. C.R. James, Univ. of Alberta, CANADA
 Dr. Peter Lukac, Komenskaho Univ, CZECHOSLOVAKIA
 The Librarian, Culham Laboratory, ENGLAND
 The Librarian, Rutherford Appleton Laboratory, ENGLAND
 Mrs. S.A. Hutchinson, JET Library, ENGLAND
 C. Mouttet, Lab. de Physique des Milieux Ionises, FRANCE
 J. Radet, CEN/CADARACHE - Bat 506, FRANCE
 Univ. of Ioannina, Library of Physics Dept. GREECE
 Dr. Tom Mui, Academy Bibliographic Ser., HONG KONG
 Preprint Library, Hungarian Academy of Sciences, HUNGARY
 Dr. B. Dasgupta, Saha Inst of Nucl. Phys., INDIA
 Dr. P. Kaw, Institute for Plasma Research, INDIA
 Dr. Philip Rosenau, Israel Inst. Tech, ISRAEL
 Librarian, Int'l Ctr Theo Phys, ITALY
 Prof. G. Rostagni, Univ Di Padova, ITALY
 Miss Clelia De Palo, Assoc EURATOM-ENEA, ITALY
 Biblioteca, Instituto di Fisica del Plasma, ITALY
 Dr. H. Yamato, Toshiba Res & Dev, JAPAN
 Prof. I. Kawakami, Atomic Energy Res. Institute, JAPAN
 Prof. Kyoji Nishikawa, Univ of Hiroshima, JAPAN
 Direc. Dept. Large Tokamak Res. JAERI, JAPAN
 Prof. Satoshi Itoh, Kyushu University, JAPAN
 Research Info Center, Nagoya University, JAPAN
 Prof. S. Tanaka, Kyoto University, JAPAN
 Library, Kyoto University, JAPAN
 Prof. Nobuyuki Inoue, University of Tokyo, JAPAN
 S. Mori, JAERI, JAPAN
 Librarian, Korea Advanced Energy Res. Institute, KOREA
 Prof. D.I. Choi, Adv. Inst Sci & Tech, KOREA
 Prof. B.S. Liley, University of Waikato, NEW ZEALAND
 Institute of Plasma Physics, PEOPLE'S REPUBLIC OF CHINA
 Librarian, Institute of Phys., PEOPLE'S REPUBLIC OF CHINA
 Library, Tsing Hua University, PEOPLE'S REPUBLIC OF CHINA
 Z. Li, Southwest Inst. Physics, PEOPLE'S REPUBLIC OF CHINA
 Prof. J.A.C. Cabral, Inst Superior Tecnico, PORTUGAL
 Dr. Octavian Petrus, AL I CUZA University, ROMANIA
 Dr. Johan de Villiers, Fusion Studies, AEC, SO AFRICA
 Prof. M.A. Hellberg, University of Natal, SO AFRICA
 C.I.E.M.A.T., Fusion Div. Library, SPAIN
 Dr. Lennart Stenflo, University of UMEA, SWEDEN
 Library, Royal Inst Tech, SWEDEN
 Prof. Hans Wilhelmson, Chalmers Univ Tech, SWEDEN
 Centre Phys des Plasmas, Ecole Polytech Fed, SWITZERLAND
 Bibliothek, Fom-Inst Voor Plasma-Fysica, THE NETHERLANDS
 Dr. D.D. Ryutov, Siberian Acad Sci, USSR
 Dr. G.A. Eliseev, Kurchatov Institute, USSR
 Dr. V.A. Glukhikh, Inst Electrophysical Apparatus, USSR
 Dr. V.T. Tolok, Inst. Phys. Tech. USSR
 Dr. L.M. Kovrizhnykh, Institute Gen. Physics, USSR
 Nuclear Res. Establishment, Julich Ltd., W. GERMANY
 Bibliothek, Inst. Fur Plasmaforschung, W. GERMANY
 Dr. K. Schindler, Ruhr Universitat Bochum, W. GERMANY
 ASDEX Reading Rm, IPP/Max-Planck-Institut fur
 Plasmaphysik, W. GERMANY
 Librarian, Max-Planck Institut, W. GERMANY
 Prof. R.K. Janev, Inst Phys, YUGOSLAVIA