

WHOI-92-42

C.2

**Woods Hole  
Oceanographic  
Institution**



---

**A System for Shipboard Analysis of Buoy Data**

by

Kenneth E. Prada

November 1992

**Technical Report**

Funding was provided by the Office of Naval Research  
under Grant No. N000-14-90-J-1490.

Approved for public release; distribution unlimited.

---

DOCUMENT  
LIBRARY

Woods Hole Oceanographic  
Institution

**WHOI-92-42**

**A System for Shipboard Analysis of Buoy Data**

by

**Kenneth E. Prada**

**Woods Hole Oceanographic Institution  
Woods Hole, Massachusetts 02543**

November 1992

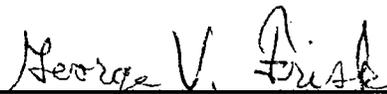
**Technical Report**

Funding was provided by the Office of Naval Research  
under Grant No. N000-14-90-J-1490.

Reproduction in whole or in part is permitted for any purpose of the United States  
Government. This report should be cited as Woods Hole Oceanog. Inst. Tech. Rept.,  
WHOI-92-42.

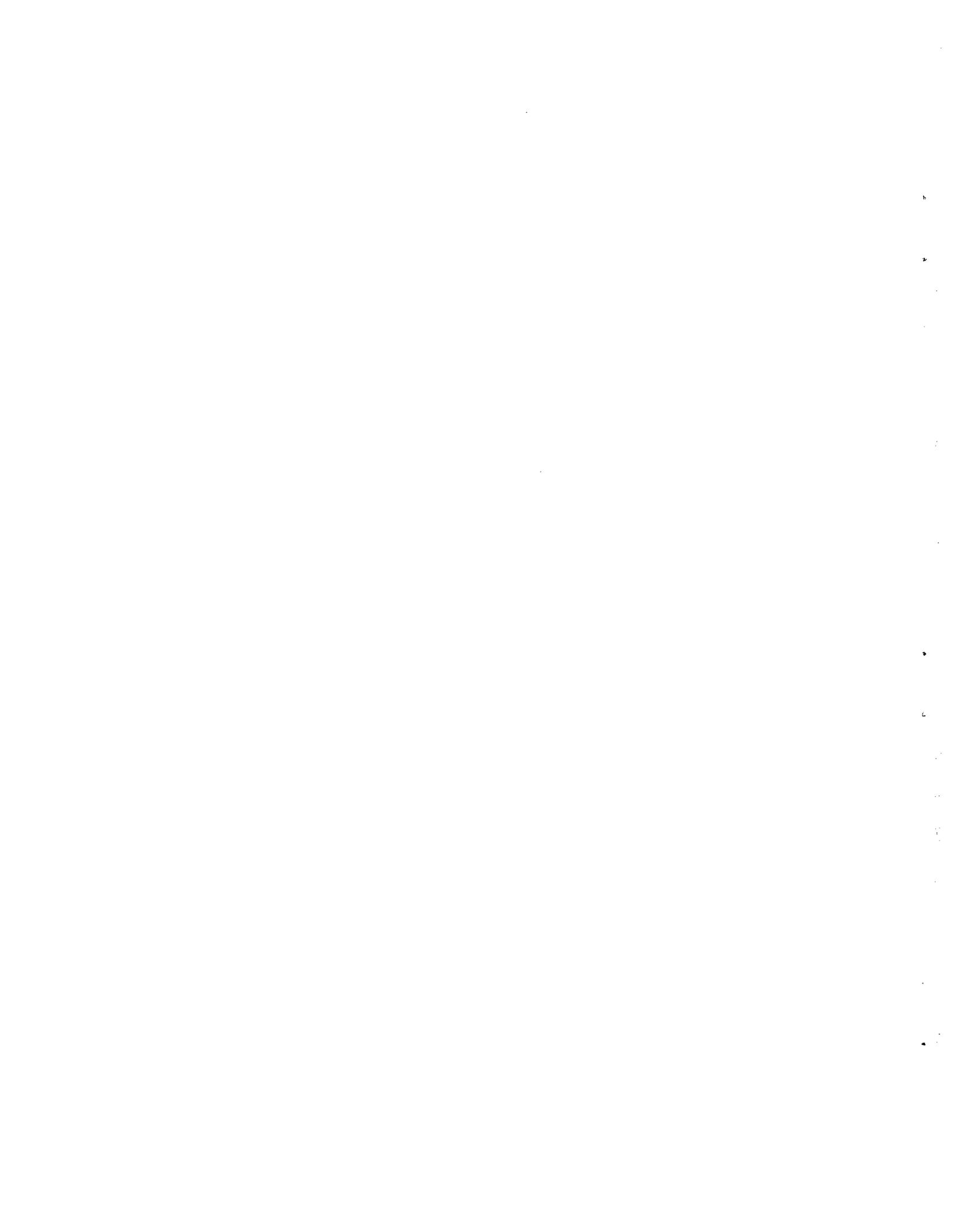
Approved for public release; distribution unlimited.

**Approved for Distribution:**



**George V. Frisk, Chairman**  
Department of Applied Ocean Physics  
and Engineering





# Contents

<b>ABSTRACT</b>	<b>3</b>
<b>1 Introduction</b>	<b>3</b>
<b>2 Data Format - UNIDATA netCDF</b>	<b>4</b>
<b>3 Shipboard Data Handling</b>	<b>5</b>
<b>4 System Structure</b>	<b>6</b>
4.1 The Primary Shell Script . . . . .	6
4.2 The UOP Menu . . . . .	6
4.2.1 X-Windows Menu Program <i>xuop</i> . . . . .	7
4.2.2 Generic Terminal Menu Program <i>tuop</i> . . . . .	7
4.3 Control and Interactive Shell Scripts . . . . .	8
4.4 Information Files . . . . .	8
4.4.1 Calibration Tables . . . . .	9
4.4.2 Plot Parameters . . . . .	9
4.5 The System Plot Package . . . . .	9
4.5.1 PostScript Files . . . . .	10
4.6 User Shell Environment . . . . .	10
<b>5 Processing Steps</b>	<b>11</b>
5.1 Extraction from Instrument . . . . .	11
5.2 Transfer to the UNIX System . . . . .	12
5.3 Conversion to netCDF . . . . .	12
5.4 Conversion to Calibrated Form . . . . .	13
5.5 Visualization Functions . . . . .	13
5.5.1 IMET Plots . . . . .	13
5.5.2 VAWR and VMCM Plots . . . . .	13
5.5.3 TPOD Plots . . . . .	13
5.5.4 Other Plot Functions . . . . .	14
<b>6 Other System Functions</b>	<b>14</b>

6.1	Calibration Table Generation . . . . .	14
6.2	Plot Parameter File Handling . . . . .	14
6.3	Environment Variable Handling . . . . .	14
6.4	Notebook Functions . . . . .	15
<b>7</b>	<b>Results and Future Work</b>	<b>15</b>
<b>8</b>	<b>Acknowledgements</b>	<b>15</b>
<b>A</b>	<b>Menu Example</b>	<b>17</b>
<b>B</b>	<b>Script, Program and Information Files</b>	<b>19</b>
<b>C</b>	<b>Plot Examples</b>	<b>21</b>

## List of Figures

1	Processing System Block Diagram . . . . .	5
2	Flow Diagram from Instrument to Data Pool . . . . .	6
3	Example of X-Windows Menu Display . . . . .	7
4	Example of Terminal Menu Display . . . . .	8
5	Processing System Flow Diagram . . . . .	11
6	IMET Plot of Seven Variables for One Day . . . . .	22
7	IMET Plot of Seven Variables for Three Days . . . . .	23
8	VAWR Plot of Seven Variables for Three Days . . . . .	24
9	VAWR Plot of Seven Variables for One Month . . . . .	25
10	VMCM Plot of Three Variables for Three Months . . . . .	26
11	TPOD Plot of One Variable for Three Months . . . . .	27
12	VAWR/IMET Overplot of Five Variables for Three Days . . . . .	28
13	Overplot of Air and Sea Temperature from Four Instruments . . . . .	29

## List of Tables

1	System Supported Plot Devices . . . . .	9
2	Some Typical System Environment Variables . . . . .	10

# ABSTRACT

This report describes a software system for shipboard processing of buoy data. These data are derived from a variety of instruments installed on buoys recovered during turn-around cruises for the SUBDUCTION experiment. The software package, which operates on a network of PCs and SUN workstations, provides an interactive and versatile set of tools for handling these data sets.

The data are extracted from the native instrument storage media and format, converted to a standard storage format in calibrated scientific units. The resulting files are then processed and displayed for evaluation and analysis. These steps occur within a short period of time following buoy recovery.

## 1 Introduction

The Upper Ocean Processes group of the Department of Physical Oceanography uses a large suite of functions for processing and analysis of buoy data. Typically these functions are performed on the Institution's central computing facilities. These functions are a group of programs created, modified, and refined over the course of many years by many individuals. Their principal advantages involve long-term familiarity and expectation of results. However, their disadvantages include a lack of versatility and adaptation, and a dependence on the central facilities architecture that does not support at-sea use.

The SUBDUCTION project needed the ability to perform processing and analysis of buoy data during recovery and turn-around cruises. The project includes up to five moorings with several eight-month deployments. Data from buoy instruments must be examined during the recovery and re-deployment cruises. Evaluations of these data would determine data integrity and indicate which instruments might be re-deployed after servicing.

Each mooring can contain up to twenty instruments with some instruments using more than eight sensors. For deployments lasting eight months a large volume of data is collected. A substantial portion of these data must be processed, displayed and evaluated in the period between mooring sites.

To support these needs, the buoy processing functions have been collected, revised, expanded, and unified into a comprehensive data processing, display, and analysis system. The system handles data from a large instrument suite, including the VAWR, VMCM, IMET module systems, and Brancker temperature pods. Other instruments, such as the XBT and ADCP, may be added in future configurations.

The system design goals are numerous and include:

- reduce dependence on VAX/VMS systems
- implement UNIX to aid long term platform portability
- implement standards for data storage formats
- implement standards for display and print graphics
- support both shipboard and shore-based laboratory needs
- simplify operations
- create and/or improve easy user interaction
- improve annotation and reporting
- increase productivity

- automate former manual activities
- establish documentation levels and standards
- allow dynamic growth and configuration

This report describes the processing software system written to support shipboard handling, analysis, and display of buoy data from the SUBDUCTION experiment.

## 2 Data Format - UNIDATA netCDF

Critical to any processing or analysis resource is the storage format. Data from the many buoy instruments exists in diverse forms. These forms need to be converted into a unified standard format for effective development of a data handling system. The netCDF format was selected.

NetCDF, the Unidata network Common Data Form [Rew & Davis, 1990A], [Rew & Davis,1990B], is an interface for scientific data access and a freely distributed software library that provides an implementation of the interface. The netCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data.

NetCDF data sets have several important features:

- self-describing - a netCDF file includes information about the data it contains.
- network-transparent - data is represented in a form that can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- directly accessible - any subset of the data can be accessed without reading over preceding data.
- extendible - data can be appended to a netCDF file without copying the prior data.

The netCDF support library and functions offer substantial advantages. The most important of these is platform portability. Most of the popular computing platforms and operating systems are supported by either Unidata or the large number of netCDF users.

NetCDF is useful for supporting access to diverse kinds of scientific data in heterogeneous networking environments and for writing application software that does not depend on application-specific formats.

NetCDF was developed at the Unidata Program Center. It is part of a data processing software development sponsored by the National Science Foundation's Division of Atmospheric Sciences. The program is managed by the University Corporation for Atmospheric Research, which maintains and distributes the netCDF software. It is freely available using ftp procedures on Internet. Current use involves broad ranging scientific applications and disciplines, both national and international.

Some of the important Unidata and netCDF advantages include:

- well documented [Rew 1991]
- multi-dimensional data sets (usually not available from database managers)
- FORTRAN and C language interface.
- UNIX, VMS, MSDOS, and OS/2 operating systems support.

- network transparent data representation.
- platform portability.
- long term support from NSF and UCAR.
- growing user base of applications and enhancements.
- an emerging accepted standard in multi-discipline data access.

### 3 Shipboard Data Handling

During the SUBDUCTION turn-around cruises, a specialized data handling and processing system was installed aboard ship. Figure 1 shows a diagram of that system. It contained two SUN IPC workstations and two DOS machines interconnected with a localized ethernet. The DOS machines are used to transfer the data from native instrument recording media or format to the file system on the processing computers

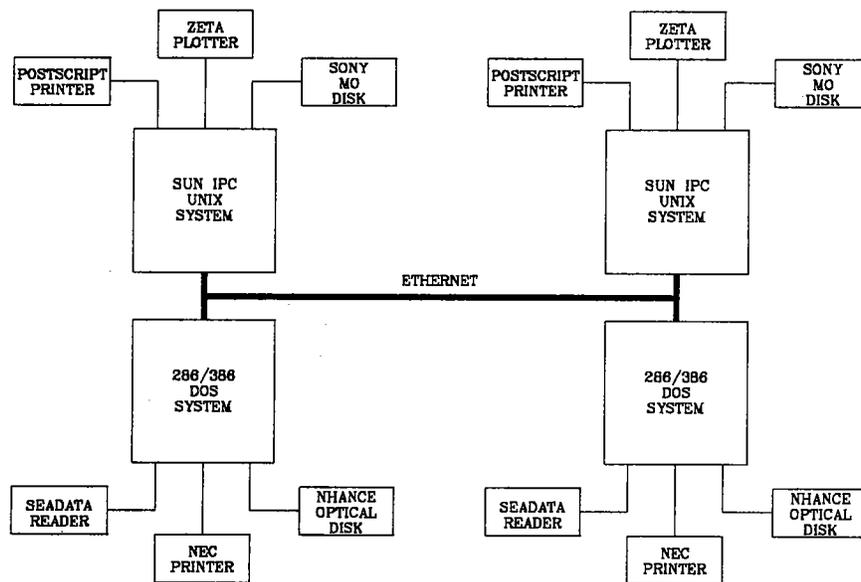


Figure 1: Processing System Block Diagram

Shipboard processing involved data from four instrument types, an IMET instrument suite, a Vector Averaging Wind Recorder (VAWR), a Vector Measuring Current Meter (VMCM), and Brancker Temperature Pods (TPOD). In all cases the data must be transferred from the instrument to the processing system. Figure 2 shows a block diagram of the data flow from each instrument to the data pool on the processing system.

The details concerning the transfer operations for each instrument type are described in detail in later sections of this report.

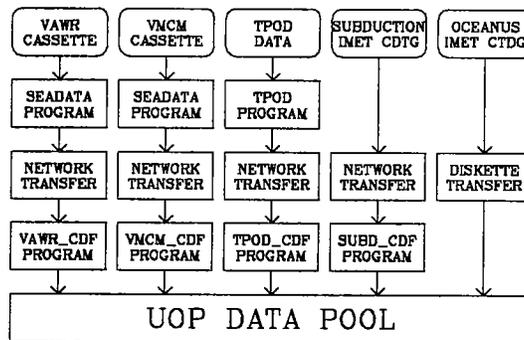


Figure 2: Flow Diagram from Instrument to Data Pool

## 4 System Structure

The system is a series of programs, shell scripts, and information files (see Appendix B). These items are combined in a unique way to allow control and execution of the complex functions that comprise the system processing and analysis. The structural elements of the system are:

1. the primary shell script `_uop`
2. menu programs `xuop` and `tuop`
3. control and interaction shell scripts
4. processing and analysis programs
5. information files

### 4.1 The Primary Shell Script

The primary shell script `_uop` is the top level control and execution element in the system structure. It interacts with the user through menu programs. The menu programs accept user requests and return functional names to the `_uop` script. The `_uop` script analyzes the functional names and executes either programs or scripts as needed. The programs or scripts return control back to the primary `_uop` script on completion.

### 4.2 The UOP Menu

The UOP menu is a dynamic object, defined in a text file `uop.mnu`. The file contains lines that define operations and name the programs that perform those operations. Lines may be organized within the file to create and name groups of similar or related functions. Appendix A contains a listing of a typical menu file.

The menu file consists of groups of functions. Each group starts with a line describing the group. Additional lines contain descriptions of each function and its terminal and X-window implementations.

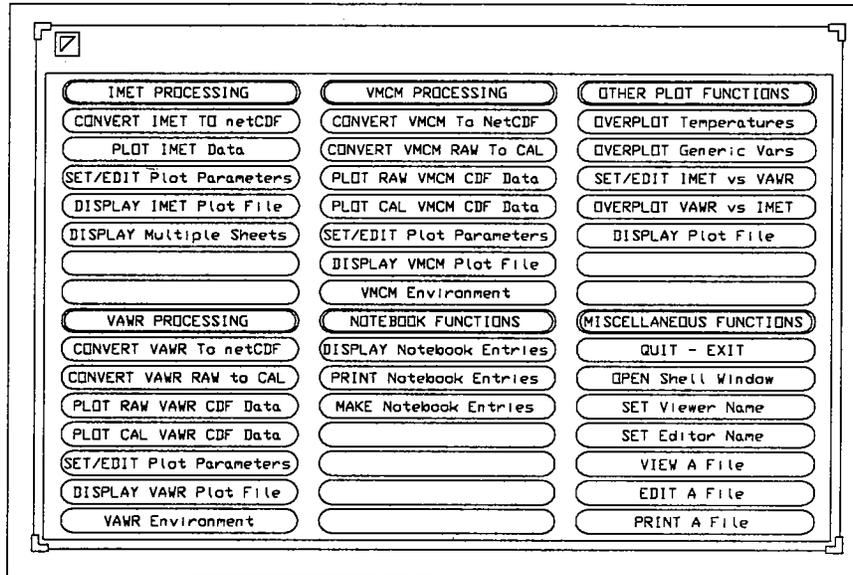


Figure 3: Example of X-Windows Menu Display

As functions are added to the system, the menu file can be revised to accommodate changing needs. Groups can be added, re-organized or removed as needs dictate. Functions can be added to groups. The system uses the menu file as a dynamic user interface to the processing functions.

#### 4.2.1 X-Windows Menu Program *ruop*

A menu program *ruop* supports the X-windows environment. This program provides a point-and-click interface to the system.

The menu definition file *uop.mnu* is used to define a window and its contents. The window contains groups of pushbuttons that describe the functions available in the system. Each group has a label button and function buttons. The label button (colored grey) contains the group name. The function buttons, all the same color within a group, contain descriptions of the functions they represent. Each group has different color buttons.

Group structure is organized for ease of viewing and comprehension. Groups are arranged in rows and columns. The number of rows and columns increase as the number of menu groups and items increase. The group with the largest number of functions determines the group size for all groups (some groups may have blank buttons). This organizes the groups so that all label buttons are aligned horizontally and groups begin and end uniformly at the top and bottom of the menu window. One button in the window is uniquely colored. This is the QUIT button. It is always colored white.

Figure 3 contains a diagram of a typical X-window system menu (without colors). This diagram should be used as a format example only, since the content may not agree with current menu files.

#### 4.2.2 Generic Terminal Menu Program *tuop*

A program *tuop* exists for system operation with terminals other than X-windows types. The program supports *termcap* operations for most terminals.

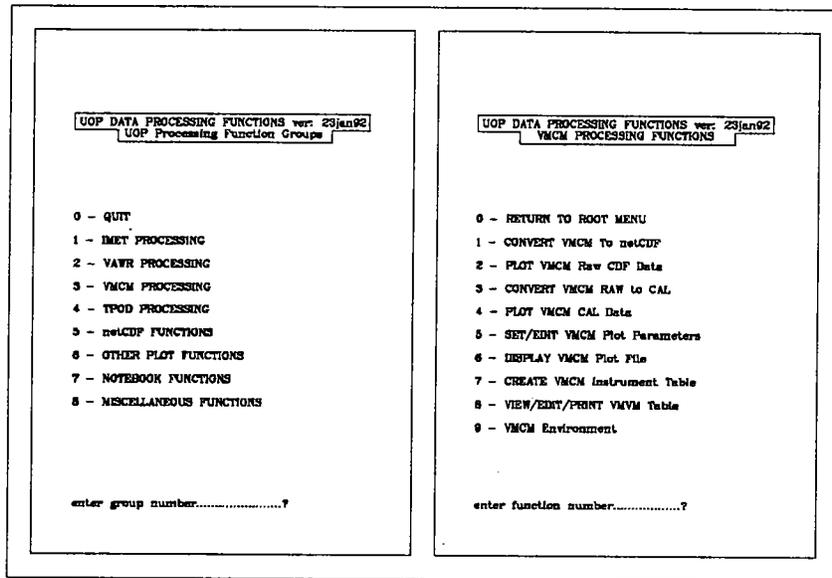


Figure 4: Example of Terminal Menu Display

Because of restricted screen size, the system menu is not displayed in its entirety on a single screen. Instead, the primary menu screen contains a list of the functional groups. The user can select a group number which then displays the contents of a that menu group. Figure 4 shows two examples of terminal menu screens.

### 4.3 Control and Interactive Shell Scripts

Shell scripts are used by the system to handle data flow, provide a batching environment, and to make the user interface simple and effective. For example, graphical output from most programs can be directed to any of several devices. The process would normally involve entering several command lines to accomplish the entire process. Instead, a shell script is interposed between the user and the several programs necessary to accomplish the task.

Information is gathered from the user by the script interactively. This might include instrument number, inclusive dates for processing, output device for graphics, or other information pertinent to the processing operation. This information is used by the script to set environment variables, call the proper programs, select output devices, and determine the order of processing steps.

### 4.4 Information Files

Several types of information files are used to enhance the processing flow. These files provide information to configure program operation thus eliminating the need for a user to enter this information at each execution instance of a program. There are also programs in the system suite specifically to create and edit these information files.

DEVICE	SIZE	PROGRAM
PostScript Device	8.5" X 11.0" (A)	<i>post11</i>
PostScript Device	8.5" X 14.0" (LEGAL)	<i>post14</i>
X11 Window		<i>xplot</i>
Tektronix 4010/4014		<i>tek (tek4014)</i>
HP 7475	11.0" X 17.0" (B)	<i>hp7475b</i>
HP 7475	8.5" X 11.0" (A)	<i>hp7475a</i>
HP 7470	8.5" X 11.0" (A)	<i>hp7470</i>
HP DRAFTPPO (7570A)	17.0" X 22.0" (C)	<i>draft_c</i>
HP DRAFTPPO (7570A)	22.0" X 34.0" (D)	<i>draft_d</i>
ZETA (HPGL)	11.0" X 17.0" (B)	<i>hp7475b</i>
ZETA (HPGL)	8.5" X 11.0" (A)	<i>hp7475a</i>

Table 1: System Supported Plot Devices

#### 4.4.1 Calibration Tables

Several instruments must have the recorded values converted into calibrated scientific units. The calibration factors are instrument and sensor specific. A table file for each instrument is created by a special program and contains calibration and scaling values for each sensor that the instrument contains. The values are derived from sensor calibration procedures performed prior to instrument use and deployment.

#### 4.4.2 Plot Parameters

There is a wide range of visualization needs for each instrument and data type. These needs must be provided for without maintaining an immense suite of specifically tailored plot programs. They must also be available without requiring the user to repeatedly enter information and parameters with each execution of a program.

Plot parameter files are used to make the visualization process a simple one. Each plot parameter file contains information concerning the desired variable(s) and scaling for each type of plot operation. The visualization programs read these files. The files may be edited either with the programs provided within the system package or with any text editor.

### 4.5 The System Plot Package

System plotting uses a generic UNIX plot method. The system is simple and lacks frills, but it is easy to use and supports a large variety of output devices (see Table 1). Plot program output is in device independent form and is directed to the *stdout* device. This allows re-direction to files for intermediate storage or piping to conversion filters

Typical plots are scaled with a 0.75:1.0 aspect ratio. This allows the use of standard paper sizes, which have approximately that ratio. Each of the device drivers scales the plot data to fit correctly with the individual device limits.

VAWRSOURCE	VAWR data input directory path
VAWRDEST	VAWR data output directory path
VAWRNOTES	VAWR notebook directory path
VAWRTABLE	VAWR calibration table directory path
VAWRNUM	VAWR instrument number for processing
IMETSOURCE	IMET data input directory path
IMETYEAR	year for IMET processing start
IMETMONTH	month for IMET processing start
IMETDAY	day for IMET processing start

Table 2: Some Typical System Environment Variables

#### 4.5.1 PostScript Files

PostScript output from the system plot functions can be handled in several ways. Output may be directed to the laser printer in two sizes, standard letter size (8.5" X 11.0") or legal size (8.5" X 14.0").<sup>1</sup> Alternatively, PostScript files may be previewed on the screen using the SUN supplied *pageview* function or the GhostScript resource from the Free Software Foundation.

#### 4.6 User Shell Environment

A user shell has an environment which consists of a number of variables (strings). These variables can convey pre-defined information to programs and scripts. An example is the environment variable *HOME* which is set by the system when the shell is initiated. *HOME* contains the name of the user's home directory (for example */usr/ken*).

The system uses environment variables to simplify its operations. They are used to signify directory paths, instrument numbers, and other information. Repetitive queries to the user are eliminated when certain variables are set by the user. The programs use the information provided in the environment variable rather than query the user.

Table 2 defines some of the environment variables used by the system. There are menu functions included for the display, setting and editing of environment variables.

The names of the system environment variables can be found in files specific to each instrument type:

```

imet.env  IMET variable names
vawr.env  VAWR variable names
vmcm.env  VMCM variable names
tpod.env  TPOD variable names

```

These are text files and contain single line entries each of which contains the name of a variable. Additions and deletions can be made to accomodate system changes and growth. This can be done with a text editor.

<sup>1</sup>Laser printer must be capable of and configured for legal size paper.

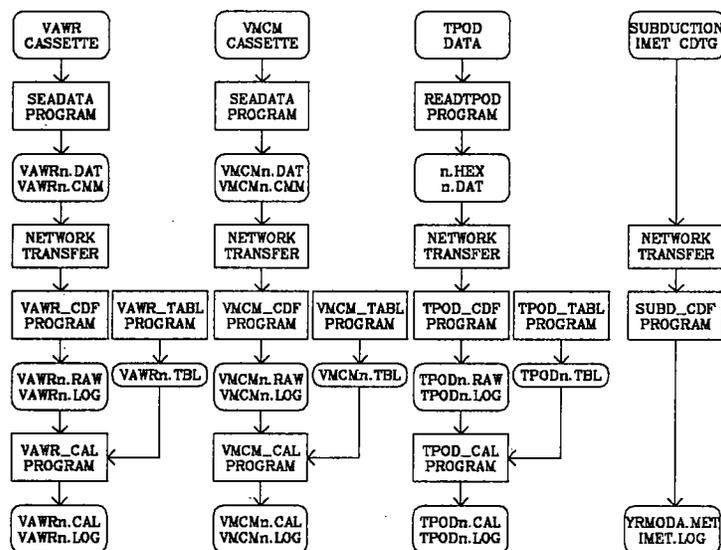


Figure 5: Processing System Flow Diagram

## 5 Processing Steps

There are several steps in the shipboard processing flow. Figure 5 shows an overall flow diagram for all instruments in the SUBDUCTION suite. The following sections describe the steps related to the individual instruments.

The shipboard system primarily performs format conversion, calibration and visualization operations. Processing of a more complex nature is not intended for the shipboard environment at this time. This report does not include information about these operations.

### 5.1 Extraction from Instrument

IMET files derived from SUBDUCTION buoys are in the original IMET format [Prada 1990]. They are typically contained on an NHance/ISI optical disk cartridge. There are 24 one-hour files per day. These can be directly transferred to the UNIX system.

VAWR and VMCM data is recovered from the instruments on a digital tape cassette. These cassettes must be processed on a DOS system. The program used to read a SEADATA cassette and create a DOS disk file is *seadata*. [Danforth 1990] This program reads the cassette and produces two files, a data file and a comment file. These files can be transferred to the UNIX system.

TPOD data is stored using internal instrument memory. A program operating on the DOS system extracts these data and creates an ASCII file with annotation and descriptive information. This file can be transferred to the UNIX system.

## 5.2 Transfer to the UNIX System

There are two transfer methods available. The first method involves the use of the PC-NFS protocol. PC-NFS is a systems protocol that mounts a disk directory on the SUN computer as though it were a disk actually resident on the DOS system. For example, the */puli/data*<sup>2</sup> directory on the UNIX system might be linked as the F: drive on the DOS system. When this method is used, transfer is accomplished by copying the desired files between the optical disk drive (typically E:) and the UNIX PC-NFS mounted drive (typically F:).

The alternative method for file transfer uses the *ftp* protocol. This method is more user intensive and would only be used if PC-NFS fails.

The IMET copy operation is done in multiple steps, since it is not possible to mount all the needed files on the DOS optical drive at the same time. There is a DOS procedure that creates a DOS script file to accomplish this task without operator intervention. The user runs a DOS program *imetzfer* that requests drive designations for the DOS optical drive and the PC-NFS drive along with start and end dates. The program then generates a DOS script file to mount and copy the required files in reasonably sized groups. This method can be implemented by typing *uop* at the DOS prompt. This executes a script which runs the *imetzfer* program and then executes the script file produced by *imetzfer*.

Copy procedures for VAWR, VMCM, and TPOD are much simpler. Since there are few files (typically two per instrument), they can be copied in a single step. These operations are typically carried out by a DOS script file.

## 5.3 Conversion to netCDF

The next step in processing for each data type is conversion to netCDF format. The IMET process request starts and end dates. One netCDF file is created for each day using 24 of the original IMET format one-hour files.

Since original IMET recording formats differ slightly from experiment to experiment, there are separate conversion programs for different experiments (some differences also exist between the deployments in SUBDUCTION). The conversion script handles the experiment differences through user queries.

File names for IMET data sets follow a chronological pattern. The original format files are named *yymmddhh.met* and the netCDF format files are named *yymmdd.met*.

VAWR and VMCM conversion to netCDF form is more complex. Since the binary files derived from the cassettes often contain data and clock errors, the conversion program must handle these issues. In most cases the error records are deleted from the resulting netCDF file. Transcripts of the error messages are displayed to the user and written to a log file (see subsequent section on notebook functions). At this stage the netCDF files only contain raw instrument counts and must undergo further processing before proper visualization can be performed.

TPOD data also is converted to raw format netCDF files. This conversion process is less demanding than the VAWR/VMCM process, since the instrument recording retrieval methods are substantially more reliable.

File names for VAWR, VMCM or TPOD data conform to a prescribed standard. That standard is: *vawrnnnn*, *vmcmnnnn*, or *tpodnnnn*, where *nnnn* is a four-digit integer representing the instrument number. This format is used throughout the system.

---

<sup>2</sup>*puli* is a SUN-IPC machine network name for one of the shipboard systems

## 5.4 Conversion to Calibrated Form

Data values from IMET instruments are already in calibrated scientific units. One calibration step is performed that corrects for magnetic compass variation at the site. This variation affects IMET wind values.

The data values acquired from the VAWR, VMCM, and TPOD instruments are not in calibrated scientific units. There are procedures for each data type to convert from raw to calibrated form. These functions read calibration information from files specific to each individual instrument and convert the raw counts to scientific units. The results are stored in second stage netCDF files that contain both the derived scientific values and the calibration information used for the derivations.

## 5.5 Visualization Functions

Visualization functions involve the display of data values in graphic form to one of several devices. Typically a series of plots will be displayed to the console screen. This can be an iterative process until the data, scaling, and annotation are acceptable. Then the resulting plot is output to a hard copy device. This is typically a PostScript device but may also be a plotter.

Appendix C contains several example plots from the SUBDUCTION instrument suite. Refer to these examples for the following sections.

### 5.5.1 IMET Plots

There are two choices for plotting IMET data. The first produces sheets or screen displays that contain a single day of measurements using the one-minute sample interval typical for IMET (figure 6). The second method plots multiple days of IMET measurements on a single sheet and may include from several days to several months (figure 7). To match the lower sampling rate for comparison to the VAWR, the data may be averaged at up to 15 minute intervals. In each case the plot contains from one to twelve trace panels within a single display frame. Each panel is labeled with minimum and maximum values and the variable name.

IMET measurements may be also plotted with other instrument measurements in overplot functions described in subsequent sections.

### 5.5.2 VAWR and VMCM Plots

VAWR and VMCM plots are variable length time series. The plots can contain up to twelve trace panels within the display frame. A VAWR plot typically has seven or eight trace panels. A VMCM plot will usually have three trace panels. The contents of each panel is user selectable (by configuration file). Minimum and maximum y-axis values may be derived from user specification or auto scaling. Figures 8 and 9 show short and long duration VAWR plots. The short duration plot is for the same period as that shown in the IMET multiple day plot. Figure 10 shows a three month VMCM plot.

### 5.5.3 TPOD Plots

TPOD plots typically contain a single trace for temperature. Each plot is a time series and may range from several days to several months. Figure 11 is a three month TPOD plot.

#### 5.5.4 Other Plot Functions

IMET/VAWR comparison plots are available. These plots contain several trace panels that display the same variable from each instrument for the identical period of time. The IMET data is usually averaged over the operational sample interval of the VAWR. The plots aid in evaluation of sensor and calibration differences during companion deployments of both instrument types. Figure 12 is an example of this plot type with five variables compared for a three day period.

Temperature overplots are also possible. This function allows multiple trace plotting in a single frame. Comparisons of air and sea temperatures from various instruments can be accomplished. The plot can handle up to twelve traces. Figure 13 shows a temperature overplot.

Comparison of various dis-similar measurements is possible with a generic overplot procedure. Similar to the temperature overplot, several traces are plotted in the same frame.

## 6 Other System Functions

The system has some basic housekeeping and maintenance functions. These provide:

- Calibration table generation
- Plot parameter creation and editing
- Environment Variable Handling
- Notebook functions

### 6.1 Calibration Table Generation

Each instrument that requires a data calibration step (VAWR, VMCM, and TPOD) have functions that create calibration information files for individual instruments. The functions are interactive, requesting value input from the user in response to specific queries. The table files thus created are used by the raw-to-calibrated conversion programs.

### 6.2 Plot Parameter File Handling

Visualization functions are controlled through the use of plot parameter files. These files contain lists of variable names and scaling values. The individual plot programs read these files and use the information to format the content and annotation of each plot frame.

### 6.3 Environment Variable Handling

Since environment variables are convenient to use, they must be readily available for change. There are functions contained in the system menu structure specifically for manipulating environment variables pertaining to each instrument type.

## 6.4 Notebook Functions

Record keeping throughout the handling and processing operations is important. The system contains functions that allow automatic or manual notebook entries.

Notebook files are created and allocated to individual instruments. Records pertaining to an instrument are recorded in the file assigned to that particular instrument. Notebook files are named with reference to the instrument. For example, VAWR number 707 would have a notebook file named *vawr0707.log*. The file would typically reside in the same directory as the data files for that instrument.

The notebook process generally begins early in the data handling sequence. For example, when a cassette from a VAWR or VMCM is read, a log file is created that contains comments from the operator and a record of errors during the read operation. When the data file is subsequently converted, calibrated and processed, the programs involved in these operations add entries to the log file. These entries may contain additional error information, program names, user names, and execution times.

The user can add entries to a log file at any time. These entries might contain any information pertinent to the data, its handling or display. The user may display or print the contents of any notebook file.

## 7 Results and Future Work

The system has been used aboard ship on two SUBDUCTION turn-around cruises and during the intervening time ashore. Production of quality control data plots during the cruise was effective and quick. These plots proved valuable in assessing individual instrument performance. It also provided an excellent evaluation of the entire data set for each mooring site and the whole experiment area.

The system needs further enhancements and additions. These should include a more generic approach to some of the features and functions, particularly visualization. Also, additional processing functions should be added to the package.

The success of this processing system has prompted a look at how similar features might be implemented for handling other oceanographic data types. A working group is considering standards and conventions for applying netCDF to a variety of data classes in oceanographic research. An Oceanographic Analysis Resource System (OARS) has been proposed, supported by a Basic Oceanographic Analysis Toolkit (BOAT). The objective will be to simplify the use and exchange of oceanographic data and provide generic tools for manipulation, processing, and visualization.

Other systems based on the netCDF data access method provide an incentive for further development of the concept. [Denbo 1991] [Wessel & Smith, 1991]

## 8 Acknowledgements

This work was supported by the Office of Naval Research under grant N000-14-90-J-1490. Maintenance and enhancement programming support has been provided by Nancy Galbraith, who with Nancy Penninton, gave valuable feedback and suggestions on operational and user interface details. Melora Park-Samelson assisted in coding most of the data calibration routines.

## References

- [Danforth 1990] Danforth, Thomas W., *The Seadata Program*, Woods Hole Oceanographic Institution Technical Report 90-44, October 1990, 73 pp.
- [Denbo 1991] D. W. Denbo, *EPIC System Library Version 2.0*, 56pp, September 1991.
- [Prada 1990] Kenneth E. Prada, *Improved Meteorological Measurements from Buoys and Ships (IMET): Preliminary Conventions and Standards, Data and File Formats*, Woods Hole Oceanographic Institution Technical Report 90-10, 11pp, March 1990.
- [Rew & Davis, 1990A] Rew, R.K. and G.P. Davis *Unidata netCDF: Software for Scientific Data Access*. Proceedings of the Sixth International Conference of Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, p 33-40.
- [Rew & Davis, 1990B] Rew, R.K. and G.P. Davis *Netcdf: An Interface for Scientific Data Access*. Computer Graphics and Applications, IEEE, pp. 76-82, July 1990.
- [Rew 1991] Russell K. Rew, *netCDF User's Guide*, An Interface for Data Access, Version 2.0, Unidata Program Center, 149pp, October 1991.
- [Rew & Emmerson, 1992] Russ Rew and Steve Emmerson, *NetCDF Operators and Utilities*, UCAR Report, 13pp, January 1992.
- [Wessel & Smith, 1991] Paul Wessel and H.F. Smith, *The GMT-SYSTEM Version 2.0 Technical Reference and Cookbook*, University of Hawaii School of Ocean and Earth Science and Technology Unpublished Document, October 1991.

## A Menu Example

This appendix contains an example of a menu file. This is an examples of format only, so content may not agree with the current menu.

```
#IMET PROCESSING
CONVERT IMET TO netCDF:_imetcdf:_imetcdf
PLOT IMET Data:_imetplt:_imetplt
IMET Plot Parameters:imet_var:imet_var
DISPLAY IMET Single Sheets:_imetshs:_imetshs
DISPLAY IMET Long Plots:_plotout:_plotout
IMET Environment:_imetenv:_imetenv
$
#VAWR PROCESSING
CONVERT VAWR TO netCDF:vawr_cdf:vawr_cdf
CONVERT VAWR netCDF RAW to CAL:vawr_cal:vawr_cal
PLOT VAWR CDF Data:_vawrplot:_vawrplot
VAWR Plot Parameters:vawr_var:vawr_var
DISPLAY VAWR Plot File:_plotout:_plotout
CREATE VAWR Instrument Table:vawr_tabl:vawr_tabl
VIEW/EDIT VAWR Table:_vawredit:_vawredit
VAWR Environment:_vawrenv:_vawrenv
$
#VMCM PROCESSING
CONVERT VMCM TO netCDF:vmcm_cdf:vmcm_cdf
CONVERT VMCM netCDF RAW to CAL:vmcm_cal:vmcm_cal
PLOT VMCM CDF Data:_vmcmplot:_vmcmplot
VMCM Plot Parameters:vmcm_var:vmcm_var
DISPLAY VMCM Plot File:_plotout:_plotout
CREATE VMCM Instrument Table:vmcm_tabl:vmcm_tabl
VIEW/EDIT VMCM Table:_vmcmedit:_vmcmedit
VMCM Environment:_vmcmenv:_vmcmenv
$
#TPOD Processing
CONVERT TPOD TO netCDF:tpod_cdf:tpod_cdf
CONVERT TPOD netCDF RAW to CAL:tpod_cal:tpod_cal
PLOT TPOD Data:_tpodplot:_tpodplot
TPOD Plot Parameters:tpod_var:tpod_var
DISPLAY TPOD Plot File:_plotout:_plotout
CREATE TPOD Instrument Table:tpod_tabl:tpod_tabl
VIEW/EDIT TPOD Table:_tpodedit:_tpodedit
TPOD Environment:_tpodenv:_tpodenv
$
#netCDF FUNCTIONS
DISPLAY netCDF File Parameters:cdf_show:cdf_show
DISPLAY netCDF File:cdf_dump:cdf_dump
EDIT netCDF file:cdf_edit:cdf_edit
PRINT netCDF File Parameters:_cdflist:_cdflist
CONVERT netCDF to ASCII:cdf_asc:cdf_asc
CONVERT netCDF to Binary:cdf_bin:cdf_bin
```

```
$
#OTHER PLOT FUNCTIONS
OVERPLOT Temperatures:_tempplot:_tempplot
OVERPLOT Generic Variables:_overplot:_overplot
SET/EDIT VAWR vs IMET Parameters:vaim_var:vaim_var
OVERPLOT VAWR vs IMET:vawr_imet:vawr_imet
DISPLAY Plot File:_plotout:_plotout
$
#NOTEBOOK FUNCTIONS
CREATE Notebook Entries:_makenotes:_makenotes
DISPLAY Notebook:_shownotes:_shownotes
PRINT Notebook:_listnotes:_listnotes
$
#MISCELLANEOUS FUNCTIONS
QUIT -. EXIT - FINIS:quit:quit
OPEN Shell Window:_dounix:_dounix
SET Viewer Name:_setview:_setview
SET Editor Name:_setedit:_setedit
VIEW A File:_view:_view
EDIT A File:_edit:_edit
OPEN UNIGRAPH Window:_unigraph:_unigraph
$
```

## B Script, Program and Information Files

This appendix contains lists of the system shell scripts and programs along with their functions.

SCRIPT NAME	FUNCTION
_cdflist	list netCDF file parameters to print device
_dounix	start a shell script
_edit	execute the assigned text editor
_imetcdf	IMET to netCDF conversion for multiple days
_imetenv	set, edit, delete IMET environment variables
_imetplt	IMET plot control, all variations
_imetshs	display control for IMET plots for multiple days
_listnotes	list notebook to print device
_makenotes	make notebook entries
_overplot	overplot control
_plotout	output control of plot files
_print	execute assigned file print process
_setedit	assign text editor process
_setview	assign text file viewer
_shownotes	screen display notebook entries
_tempplot	temperature plot generation control
_tpodcdf	TPOD to netCDF conversion control
_tpodedit	edit TPOD table file
_tpodenv	set, edit, delete TPOD environment variables
_tpodplot	TPOD plot control
_unigraph	setup and execute Unigraph process
_uop	primary UOP control script
_vawredit	edit VAWR table files
_vawrenv	set, edit, delete VAWR environment variables
_vawrimet	VAWR/IMET overplot control
_vawrplot	VAWR plot control
_vawrraw	VAWR raw data plot control
_view	execute assigned file view process
_vmcmedit	edit VMCM table files
_vmcmenv	set, edit, delete VMCM environment variables
_vmcmplot	VMCM plot control
_vmcmraw	VMCM raw data plot control
cdf_asc	convert netCDF files to ASCII
cdf_bin	convert netCDF files to binary
cdf_dump	screen display netCDF file records
cdf_edit	edit netCDF file records
cdf_list	list cdf file records to print device
cdf_scan	scan netCDF file for minimum and maximum values
cdf_show	display netCDF file header and parameters
disp_env	display environment names from selected file
echo2	output to <i>stderr</i> for script
gotoxy	screen cursor control from script

PROGRAM NAME	FUNCTION
imet_env	IMET environment variable handling
imet_log	IMET notebook entries
imet_long	IMET long plot function
imet_name	create IMET file name from environment variables
imet_nlist	print contents of IMET notebook
imet_notes	display contents of IMET notebook
imet_plot	generate a one-day IMET plot
imet_var	edit imet plot variables file
next_day	generate environment variables for next day
over_plot	generic overplot function
prev_day	generate environment variable for previous day
subd1_cdf	convert SUBDUCTION 1 data to netCDF
subd2_cdf	convert SUBDUCTION 2 data to netCDF
temp_plot	plot temperatures from various instruments
title	display a title line on the console screen
tpod_cal	convert TPOD raw data to calibrated netCDF
tpod_cdf	convert TPOD data to netCDF
tpod_env	manipulate TPOD environment variables
tpod_log	create TPOD notebook entries
tpod_nlist	print TPOD notebook entries
tpod_notes	display TPOD notebook entries
tpod_plot	plot TPOD data
tpod_tabl	generate table file for TPOD data
tpod_var	edit TPOD plot variables
tuop	terminal menu program
uopxplot	X-windows plot output with mouse interaction
vaim_var	edit VAWR/IMET plot variables
vawr_cal	convert raw VAWR to calibrated
vawr_cdf	convert VAWR data to netCDF
vawr_env	manipulate VAWR environment variables
vawr_imet	generate VAWR/IMET comparison plots
vawr_log	create VAWR notebook entries
vawr_nlist	print VAWR notebook entries
vawr_notes	display VAWR notebook entries
vawr_plot	plot VAWR data
vawr_raw	plot VAWR raw data
vawr_tabl	generate table file for VAWR data
vawr_var	edit VAWR plot variables
vmcm_cal	convert raw VMCM to calibrated
vmcm_cdf	convert VMCM data to netCDF
vmcm_env	manipulate VMCM environment variables
vmcm_log	create VMCM notebook entries
vmcm_nlist	print VMCM notebook entries
vmcm_notes	display VMCM notebook entries
vmcm_plot	plot VMCM data
vmcm_raw	plot VMCM raw data
vmcm_tabl	generate table file for VMCM data
vmcm_var	edit VMCM plot variables
xplot	X-windows plot output
xuop	X-windows menu program

## C Plot Examples

This appendix contains examples of various plots from system processing output.

- figure 6 IMET plot of seven variables for one day
- figure 7 IMET plot of seven variables for three days
- figure 8 VAWR plot of seven variables for three days
- figure 9 VAWR plot of seven variables for one month
- figure 10 VMCM plot of three variables for three months
- figure 11 TPOD plot of single variable for three months
- figure 12 VAWR/IMET overplot of five variables
- figure 13 Overplot of air and sea temperatures from four instruments

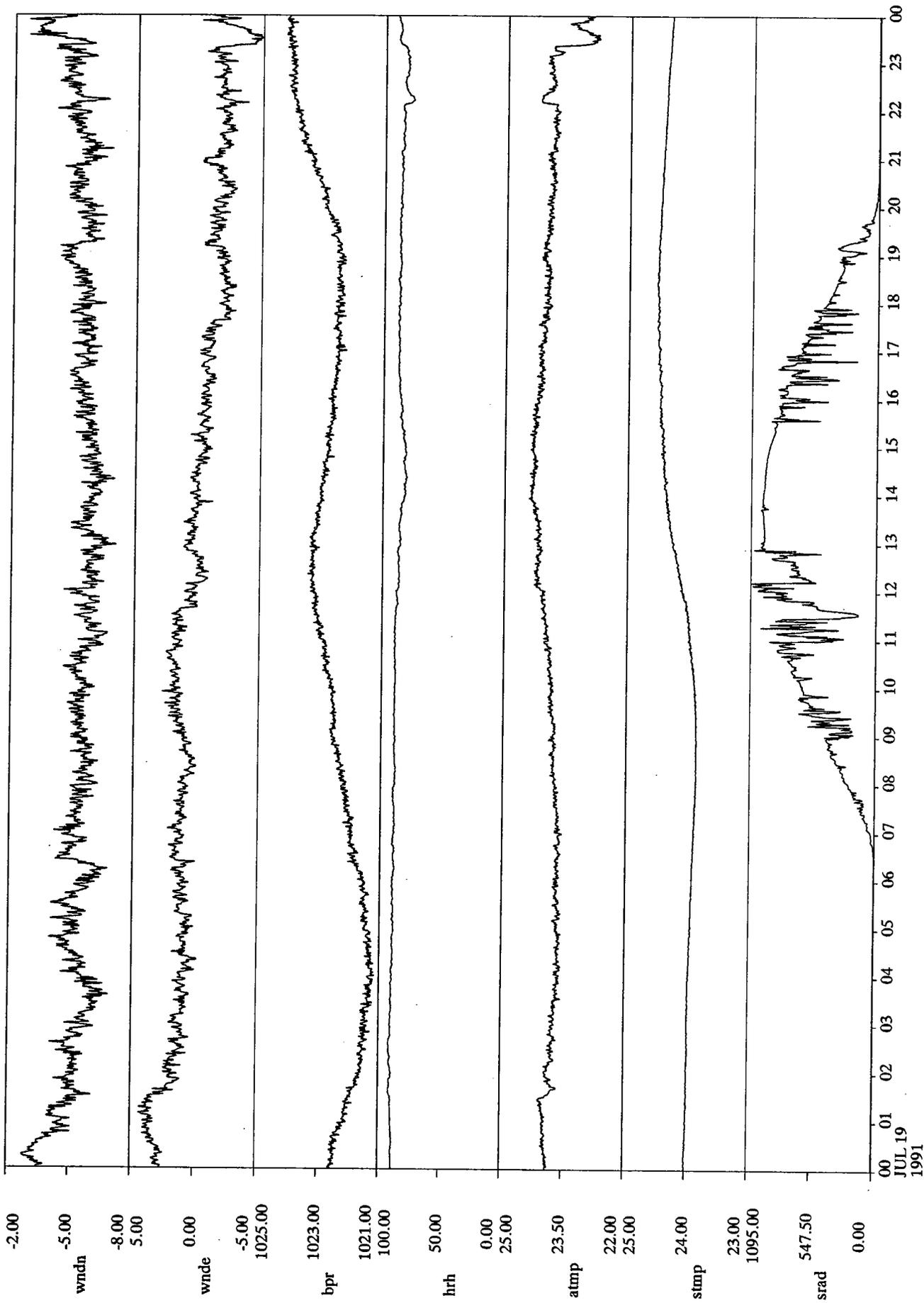


Figure 6: IMET Plot of Seven Variables for One Day

