Unkel Software Inc.
62 Bridge Street, Lexington, MA 02173

UnkelScope Documentation - What There is and How to Use it

Unkel Software provides a variety of documentation to serve users of UnkelScope who have different levels of experience and different styles of learning. This section describes the available documentation and how it is intended to be used. We welcome your suggestions and comments about the existing documentation or about additional documentation that you feel would be useful.

1. Quick Start Notes

   Intended for the user who is familiar with the PC and generally with the data acquisition board and computer data acquisition in general. The Quick Start notes show how to install the software and hardware, and how to run UnkelScope with the minimum of detail.

2. Short Guides

   The short guides summarize the features of UnkelScope and how to use them. Hardware connections are shown and references to the Users Guide are made so you can easily find full details.

3. Installation Guide

   This provides full details about the hardware and software installation, and about checkout and error identification.

4. Users Guide

   The Users Guide is the full documentation of the program features and usage. Chapters 1-6 teach the basic sampling, display and storing capabilities, while also demonstrating some of the limitations of the system. Chapter 7 discusses the manipulation of data files written with UnkelScope. Chapters 8-17 cover the extensive in-program processing and experiment control features of UnkelScope. This document gives examples of use and should be read or skimmed by all users.

5. Tutorial Disk

   The tutorial disk shows how to use the basic features of UnkelScope by showing the sampling and display of some simulated signals.

6. Release Notes

   The release notes summarize changes and additions specific to recent releases of UnkelScope. If you have received a software update you should read the Release Notes.

7. Technical Notes

   Technical Notes cover specific aspects of software and hardware use. These notes will be referenced by an appropriate section of other parts of the documentation and can be found at the end of the documentation notebook.

(617) 861-0181

UnkelScope

VERSION FOR:

IBM DACA

Release 3.05 April 1988

Unkel Software Inc.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1.0 How to Use UnkelScope Documentation

Welcome to the UnkelScope Users Guide. This Users Guide is one portion of the documentation provided to help you to learn how to use UnkelScope and your computer based data acquisition hardware efficiently and correctly. You can also use the tutorial program and/or the Short Guide.

This guide is intended to help you learn UnkelScope as you use your data acquisition hardware. The guide is divided into 3 parts, roughly corresponding to the capabilities of the three levels of UnkelScope: Level 1, Level 2 and Level 2 Plus. In this approach to documentation you must expect that some terms or details of operation will be incompletely defined at the time you use them. Completing Sections 2, 3 and 4 should allow you to use UnkelScope for routine data acquisition tasks. Sections 5 and 6 demonstrate some of the potential problems with discrete sampling systems. If you are unfamiliar with computer-based data acquisition you should read Sections 5 and 6. It may be instructive to skim the entire document to get an idea of the overall capabilities.

If you are an experienced user of computer-based data acquisition systems you may not need the User Guide except to clarify the detailed use of UnkelScope features. In this case use the Table of Contents and the Index to help you find the material you want.

If the hardware and/or software has not already been installed, or you have any questions about the hardware, you should consult the UnkelScope Installation Guide and/or the hardware vendor's documentation. Questions about the operation system, the keyboard or other things related to the computer are addressed by the computer manuals.

If you have questions about the operation that are not answered by any of the documentation provided, call Unkel Software at (617) 861-0181.

Part 1

The first part is designed to teach you how to use the system as a substitute for analog equipment. Part 1 also points out potential problems and shows you how to avoid them. The guide is organized as a sequence of short descriptions followed by either an exercise or a demonstration of results. The first exercise will show how to use the program to sample and display (on the screen and in hardcopy) waveforms and store and retrieve the data in a disk file. The second exercise demonstrates several issues of discrete time sampling thus giving you the confidence to use the software and hardware effectively. The final exercise shows how to manipulate the data off-line. The environments supported for offline analysis are FORTRAN, BASIC, RS/1 and LOTUS-123. If you have Level 2 or Level 2+ of the software you should consider whether its capabilities are sufficient to meet your needs.

The guide assumes that you already know the basic operating procedure for the personal computer, so no discussion is given of procedures such as how to turn on and "boot-up" the system. This guide assumes that the reader has a notion of the discrete time sampling process, but not necessarily a detailed understanding of the process. In Section 7, it is assumed that you are knowledgeable of the particular environment you chose for offline data reduction - i.e. Fortran or Basic.

To use Part 1 read Sections 1, 2 and 3; these are to give you the general idea of what the system does and how it is structured. Section 4 is the first exercise - only then should you begin to work directly with the computer system. Section 5 is a general discussion of discrete time sampling. After reading Section 5, read and follow the instructions of Section 6 which is the second exercise. Finally, Section 7 describes the offline processing capabilities for UnkelScope Level 1. UnkelScope Level 2 provides data processing and other functions and is an alternative to the environments described in Section 7.

Part 2

This part is designed to show the data processing and experiment control activities of UnkelScope (Level 2 and Level 2+ only.) The material is organized into sections corresponding to the selections in the software itself. In each case discussion focusses on an illustrative example. Before using any of these sections, you should read Section 8 which describes the overall approach taken and presents a summary of the activities. You might find it useful to "browse" through this section to see what is available and return to the sections later as you actually use them.

Part 3

This part describes the graphical editing and "procedure" (or macro) capability of UnkelScope (Level 2+ only.) Graphical editing includes the ability to scroll through data plotted on the screen, showing the numerical value of the currently marked point. Procedures (macros) are a way to store and re-perform a set of operations. For example, a procedure can be established to take a set of data, filter it to reduce noise and to create a hardcopy of the resulting data. Once this is done, the procedure can be performed again without re-entering each of the keystrokes.

These sections are described as a set of examples. Again, you might wish to browse this section, and then read the section in detail as you need to use it.

2.0 Basic Structure and Functions

This section and Section 3 should be read quickly as preparation for Exercise 1 of the guide. You should not start using the computer until you begin Section 4.

The UnkelScope interface has three parts:

a) the command mode;

b) the parameter setup mode; and,

c) the data sampling and display mode.

The command mode allows you to select a general task to perform (Figure 2.1a shows the choices). The most often used tasks are the parameter SETUP MODE and the data SMPL/DSPLY MODE; other tasks include storing the data, making hard copy of the data and storing and retrieving disk files that specify the parameter options of the setup screen. The parameter setup mode (Figure 2.1b shows one set of parameters) allows you to select the sampling and display parameters such as the signal source, the range of the display, the sampling interval and the number of samples. The data sampling and display mode (Figure 2.1c shows the display for one case) presents the data plot. The upper left corner is used to display messages and to make requests for special user inputs.

```
                                                                   UnkelScope (c) MIT 1984
Commands          QUIT       SETUP MODE   SMPL/DSPLY   SAVE SETUP   GET SETUP
               SAVE DATA     PRNT SETUP   PRNT DSPLY    UTILITIES
```

Figure 2.1a. Choices in the command mode.

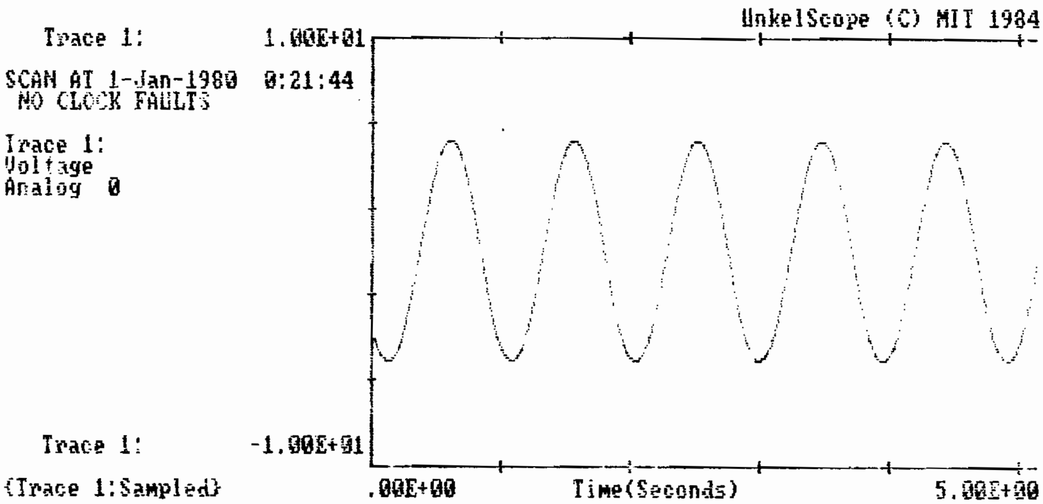


Figure 2.1c. Sample of the display screen.

```
---------- Vertical Trace 1 ----------   ------ Additional Vertical Traces ------
Source [Analog  0]  A/D Range [+/- 10]   Tr Input                            A/D
Label  :Voltage                          #  Chan  Label:                   Range
Span   [ 20    v full scale]             3 [none ]
Range  [-1.00E+01 to  1.00E+01]          4 [none ]

---------- Vertical Trace 2 ----------
Source [None      ]

                                         ------------ Triggering ------------
                                         Mode  [Auto        ]

---------- Horizontal Trace ----------   ------------ Processing ------------
Source [Time      ]                      Type [none                       ]
Label  :Time(Seconds)
Span   [ 5    s full scale]
Range  [  .00E+00 to  5.00E+00]

-------------- Sampling --------------
Sample Rate [  5 ms        200 hz]
[1024] Samples  {Scan Time 5.12E+00 s}
{Delayed Plot  } {Processing Inactive}
```

Figure 2.1b. Sample of the setup parameter screen.

3.0 Selection of Activities and Choices for Parameters

There are 3 types of interactions with the scope program through the keyboard. The following describes the "jargon" used in this guide.

Option - A list or sequence of possibilities relating to a general topic.

Choices - The actual entries in the list.

Toggling - Toggling refers to selecting from a sequence of choices. This is done by

a) moving (see entry below) to the choice and displaying it in reverse video.

b) toggling with the left arrow (previous) and/or right (next) arrow keys to the selection of interest and

c) moving from and executing the option with the current choice as discussed below. The command mode is considered an option of activities. For the command mode all choices are displayed with the current choice displayed in reverse video. For the options in the setup mode, only the current choice is displayed and toggling to the next or previous choice results in the new choice overlaying the old choice on the screen.

Executing an Option - The command "line" is considered an option. In the setup mode, each line (or part of a line) on the screen represents an option. Moving from an option "enters" the current choice of that line and initiates any auxiliary activity associated with it; this is referred to as executing an option. The up and down arrow keys are used to move to an option; in the case of the command mode this is interpreted as executing or entering the activity. In the setup mode, executing an option also moves you to the next (down arrow key) or previous (up arrow key) option. The Return key and Enter key (if there is a separate one) are interpreted the same as down arrow key.

Responding to : - Some options require character input. This is indicated by a : at the end of the option and a reverse video field that shows the maximum length of the desired input. In some cases (sample display mode and command mode) the requests for this information will be made in the upper left region of the screen. The delete key deletes one character and pressing Ctrl and U simultaneously deletes the full line of characters. The arrow keys cannot be used to 'edit' a line.

Summary - Use the up and down keys to move to a new option and to "execute" the option with the current choice. The left and right arrow keys toggle to the previous or next choice. Respond to a : by typing the appropriate information; these requests may be made in the message line located in the upper left corner of the screen or on the option line itself.

4.0 Exercise 1 - Sample, Display and Produce Hardcopy Plots

The purpose of this exercise is to show you how to use the program to acquire data and make hardcopy and screen displays of this data. When you have completed the exercise you will be able to use the system as a replacement for a recording system such as an oscilloscope or a strip chart recorder, subject to the constraint of sampling speeds and total recording time. Later exercises will show the additional features of the program.

Additional Equipment. In addition to the computer based system, you should have a source of time varying analog voltage (sinusoidal or triangle waveform) with a frequency of 10-100 Hz, and voltage level of about half the full scale input of the system peak level and an appropriate connector between the source and the analog-to-digital (A/D) input of the computer system. The example below assumes you have set up a 10 Hz triangle wave with a peak-to-peak amplitude of approximately 5 volts.

4.1 Setup of Analog Equipment to the Computer

Hook up the analog source of voltage to the first channel of the A/D system (referred to as Ch 0). Figure 4.1 shows the connections for your hardware. Turn on the voltage source, and if you have any doubts about the instrument, check it with another instrument. Set the function generator to the appropriate settings.

J4 Connector Pin Locations

(Top of Computer)

Signal Name	Pin #		Signal Name
D/A Output Channel 1	1	2	D/A Output Channel 0
+ 10 Volt Reference	3	4	Analog Ground
A/D Input Channel 0 Low	5	6	Analog Input Channel 0 High
A/D Input Channel 1 Low	7	8	Analog Input Channel 1 High
A/D Input Channel 2 Low	9	10	Analog Input Channel 2 High
A/D Input Channel 3 Low	11	12	Analog Input Channel 3 High
Analog Ground	13	14	A/D CE
Digital Ground	15	16	A/D CO
Digital Input Bit 8	17	18	Digital Output Bit 8
Digital Input Bit 9	19	20	Digital Output Bit 9
Digital Input Bit 10	21	22	Digital Output Bit 10
Digital Input Bit 11	23	24	Digital Output Bit 11
Digital Input Bit 12	25	26	Digital Output Bit 12
Digital Input Bit 13	27	28	Digital Output Bit 13
Digital Input Bit 14	29	30	Digital Output Bit 14
Digital Input Bit 15	31	32	Digital Output Bit 15
Digital Input Hold	33	34	Digital Output Gate
Digital Input Bit 0	35	36	Digital Output Bit 0
Digital Input Bit 1	37	38	Digital Output Bit 1
Digital Input Bit 2	39	40	Digital Output Bit 2
Digital Input Bit 3	41	42	Digital Output Bit 3
Digital Input Bit 4	43	44	Ditital Output Bit 4
Digital Input Bit 5	45	46	Digital Output Bit 5
Digital Input Bit 6	47	48	Digital Output Bit 6
Digital Input Bit 7	49	50	Digital Output Bit 7
Rate Out	51	52	Delay out
Digital Input Strobe	53	54	Digital Output Strobe
Digital Output CTS	55	56	Digital Input CTS
IRQ	57	58	Counter Output
Counter Input	59	60	Digital Ground

(Bottom of Computer)

Figure 4.1. Summary of interface between the computer data acquisition board and the measurement sensors.

(This Page Intentionally Blank)

4.2 Run the UnkelScope Software

If you have not started and "booted-up" the computer you should do so now. If the UnkelScope software resides on a hard disk, read and follow the instructions of Section 4.2.1, otherwise read Section 4.2.2. These sections assume that the instructions in Section 4 of the UnkelScope Installation Guide have been completed.

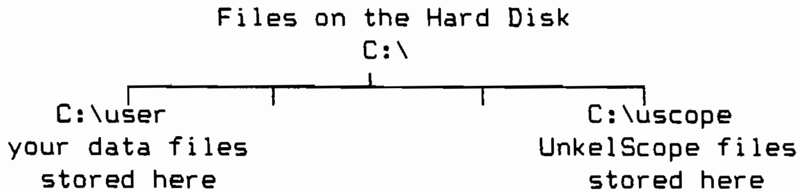
4.2.1 Hard Disk System

Move to the subdirectory in which you wish your data files to be stored. Start the program by typing (to the system prompt C> for the IBM PC running DOS)

SCOPE

This assumes that the instructions of Section 4 of the UnkelScope Installation Guide have been completed such that the UnkelScope program and associated files are located in a subdirectory c:\uscope and that the command "path=c:\uscope" has been entered. For example, if you wish to store files in subdirectory "c:\user", you should type:

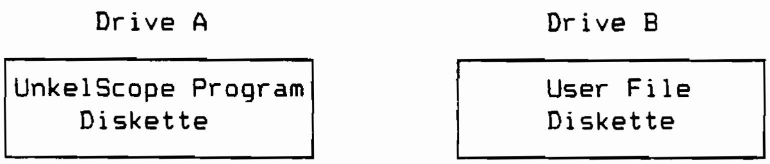
```
C> cd c:\user
C> scope
```



4.2.2 Floppy Disk System

For the floppy disk system, insert the UnkelScope Program disk into Drive A: and a formatted disk in Drive B. Identify the B drive by typing (to the system prompt) "B:". For example, if the current disk is A, the sequence of commands would be

```
A> B:
B> scope
```



Sketch of setup for using UnkelScope with a dual floppy disk system.

4.2.3 Welcome Message and Hardware Summary

The computer will read the program input file and display a welcome message and an equipment summary message. For your system the message should be as shown in Figure 4.2. If the message is different from that displayed in Figure 4.2 consult the person in charge of the system. When you have checked this, press return and you will be moved to the command mode with the current choice of setup mode.

If no messages is here, the software
may not have been properly installed.
If this is the case you should read and
follow the instructions of the UnkelScope
Installation Guide.

Figure 4.2 Welcome message and equipment summary.

Installed by ________________________.

Date ____________________.

4.3. Exiting the Program

To exit, or quit, the program "normally" you select the Quit choice in the command mode. Use the left and/or right arrow keys to toggle to the Quit choice and press the down arrow key (or up arrow key or Return) to execute this option. Since any data not previously stored in a disk file will be lost upon quitting, the program double checks whether you really want to quit by requesting (in the message region) that you type a Y (or y) and then a carriage return to confirm. To avoid having to restart the program again, press "n" (actually any character except a Y or y) and then return. You will return to the command option with the current choice of quit.

If the program is hopelessly fouled up you may wish to exit the program without returning to the command mode and executing the quit choice. If you do this, you will lose any information not stored on disk by the program and there may be no visible cursor when you return to the operating system. To do this on the IBM Personal Computer hold down the Ctrl key and type C (designated Ctrl/C). Depending where you are in the program, you may need to type more than one Ctrl/C. If you are in the sampling process you may need to type ESC before the Ctrl/C. Since this exit is issued at the operating system level, there is no chance to change your mind. With the IBM Personal Computer you can also exit abnormally and "reboot" the system by pressing Ctrl,Alt, and Del keys simultaneously, or by turning off the system console. As with Ctrl/C these actions destroy any data not previously stored in a data file.

4.4 Setup Mode

Toggle to the SETUP MODE choice using the left or right arrow keys. The setup parameters are organized into logical blocks of information roughly equivalent to the organization of an analog instrument such as an oscilloscope or stripchart recorder. These blocks are:

a) Vertical Trace Parameters, one block for each of the 2 Vertical Traces displayed and a third block for the additional channels sampled but not displayed.
b) Horizontal Trace Parameters;
c) Sampling Parameters; and,
d) Triggering and Repetition Parameters.

For this exercise, you will need to modify parameters in the Vertical Trace 1 block, the Horizontal Trace block and the Sampling Parameter block. Figure 4.2 shows the order in which these are set and an image of how the parameters might be set when you are done with this section. The meaning of each of the options and choices is described in the sections below.

4.4.1 Vertical Trace 1 Block

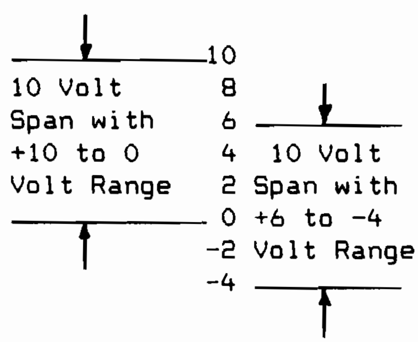
With the choice Setup Mode shown in reverse video, press the down arrow key. The cursor will display the header and the first option line of the Vertical Trace 1 block.

Source Option. The default for this is the Analog Channel 0, so you needn't modify it. Move to the next option which is the A/D Range Option.

A/D Range Option. If your A/D converter has a software selectable gain, the A/D Range option will allow you to select the appropriate gain. In this case the current range choice will be shown and you can toggle to the appropriate range using the the arrow keys. If your A/D converter has no software programmable gain the actual range of the A/D converter will be shown; you will not be able to toggle to other choices. For this exercise you do not need to alter this option.

Label Option. A 20 character label can be input to identify the trace. This label appears on the plot and with the disk file if the data is stored. The : indicates that alphanumeric characters are expected and the reverse video region indicates the 20 character length allowed. The "delete" key deletes one character, Ctrl/U deletes the whole line. Modify the label to read "triangle input" (or "sinewave input") and execute the option by pressing the down arrow key (or the return key).

Span and Range. Together the span and range define the minimum and maximum of the displayed signal. The span defines the difference between the maximum and minimum while the range moves the "window" relative to zero. The detailed operation of each of these controls is described below.



PROCESSED INPUTS TO TRACE 1 AND TRACE 2. <u>Release 2.25 and Later</u>

With Release 2.25 and later it is possible to "process" values in real time and store and plot these values in Traces 1 and 2 of UnkelScope. This option is selected by toggling to the "[Proces]" option for Trace 1 (or 2) and pressing the down arrow key. Then set up the caluculation by toggling to the desired values for 3 additional fields. For example, to subtract the signal that goes into Trace 3 from the signal that goes into Trace 4, and the setup would be as follows:

```
                   "Operand 1"
                    |
                    v
Output    = [Tr 4] [ - ] [Tr 3]   <-"Operand 2"
                     ^
                     |
                    "Operation"
```

Note that it is the Trace Number, not the Analog Input number that is being set here, and that the appropriate signals would need to be specified as the input to Traces 4 and 3. The Operations allowed include at the least: addition, subtraction and multiplication. The operands are chosen from the available vertical traces, and the value 0.0.

DIGITAL AND TIMER INPUTS. <u>Release 2.50 and Later for specific boards.</u>

For Release 2.50 and certain boards, UnkelScope supports Digital Word/Byte Input and Counter/Timer Input operations. The coverage depends upon the available hardware on the specific board, and is available only for non-DMA (lower speed) sampling situations. If available, these input options will be referred to in the Configuration program and are activated into Traces 3 through 8, by toggling to the options [Dig 0] etc. These options cannot be input directly into Traces 1 or 2, but using the Processing described above, they can be routed to either of these traces (eg. [Tr 3] [+] [0.0]). The details of operation are very hardware specific, so look to your Release Notes to see if these features are available and if they are for detailed instructions about their use.

```
(1) Vertical Trace 1
Source [Analog  0]  A/D Range [+/- 10]
Label   :Triangle Input
Span    [ 20    v full scale]
Range   [-1.00E+01 to  1.00E+01]

(3) Additional Vertical Traces
Tr Input                          A/D
#  Chan  Label:                   Range
3 [none ]
4 [none ]

(2) Vertical Trace 2
Source [None       ]

(6) Triggering
Mode   [Auto        ]

(4) Horizontal Trace
Source [Time       ]
Label   :Time(Seconds)
Span    [ 0.5 s full scale]
Range   [   .00E+00 to  5.00E-01]

(7) Processing
Type [none                       ]

(5) Sampling
Sample Rate [  1 ms        1  khz]
[ 512] Samples  (Scan Time 5.12E-01 s)
(Delayed Plot  ) (Processing Inactive)
```

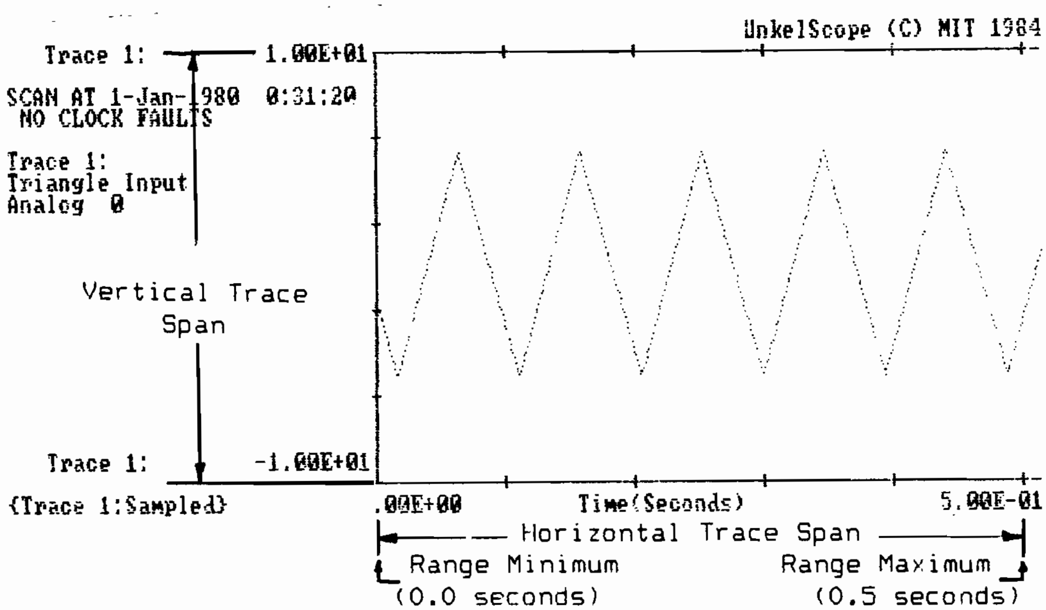


Figure 4.3. Setup screen as it might be modified for Exercise 1. Definition of span and range minimum and maximum.

Span Option. This option defines the full scale, peak-to-peak voltage range to be plotted. Select a peak-to-peak value consistent (i.e. larger than) your input by toggling to smaller spans with the left arrow key or larger spans with the right arrow key. The progression is 1, 2, 5, 10 etc. with spans from 0.1 millivolt to 1000 volts; the option "wraps around" i.e. the next choice after the highest span is the lowest span. Toggle to the value appropriate for your signal and move to the next option by pressing the down arrow key.

Range Option. The range option selects what region, at the selected span, you wish to display. The default chooses the lower limit as 0 and the upper limit as the span. For a centered analog signal, you wish 0 to be at the center of the plot. To do this you toggle the center voltage up or down by (repeatedly) using (the next) right arrow key or (previous) left arrow key. Each time you press the key the range will shift by 1/10th of the span and the range (min and max) are updated. The program allows you to offset the zero by a distance of 10 times the span. Unlike the Span Option this option does not "wrap around."

Relation of Span and Range to A/D Range and Sensitivity. The selection of the display span/range does not ensure the A/D can measure the signal accurately or even at all. For example, if the A/D Range is +10 to -10 selecting the display range from 20 volts to 30 volts will result in no data plotted. If the actual was greater than +10 volts the A/D will read its maximum value for every sample. (To help you detect when this has happened, out of scale vertical values are plotted slightly above or below the grid lines.) Similarly, if the best resolution of the A/D converter is 5 millivolts, selection of the display range -10 millivolts to +10 millivolts will result in data with a large relative digitization error. Exercise 2 (see Section 6) will demonstrate these issues in some detail. Be careful never to apply voltages above those recommended by the manufacturer.

4.4.2 Skipping Blocks

Move to the Horizontal Trace block. Specifically use the down arrow key to move through Vertical Trace 2, Additional Vertical Traces without making modifications. Since you are sampling only one channel, the inputs to these other traces will be specified as "none."

4.4.3 Horizontal Trace 1 Block

Source. The first option in the Horizontal/Time Block is the source. The default is a time base. This is what you wish so move to the next option.

Label. A 20 character label is used to identify the horizontal trace. The default when using time is "Time(seconds)". Move to the next option by pressing the down arrow key.

Span and Range. These options are similar to the display span/range options in Vertical Trace Block. The selections here define the character of the displayed data; the sampling rate is not determined by the choices of these options. Toggle to the desired settings for the span and range and move to the next option. You should display about 5 cycles of the analog signal, so for 10 Hz input signal a span of 0.5 seconds is appropriate. The default range starts at 0.0 so the range does not need to be modified.

4.4.4 Sampling Parameters Block

Sample Rate. This option defines the speed of the sampling, representing this as a time between samples and a rate (frequency) of sampling. In this case where one channel is sampled, 1 value is taken when the specified "Time Between Samples" passes. For a multiple sampling case, one sample of each sampled analog input is made as quickly as possible each time the "Time Between Samples" elapses. The sample rate toggles in 1, 2, 5, 10 increments from the smallest interval consistent with your hardware, to a time of 500 seconds between samples. Select a speed appropriate to your input i.e. to give at least 20 points in each period. For a 10 Hz signal, a delay of 1 millisecond (1 kHz) is more than adequate.

Number of Samples. The Number of Samples is toggled in factors of 2, starting at 8 and increasing to the limit consistent with your system. Toggle to get a sample number that is consistent with the Time Between Samples and with the displayed range of the time sweep. For a time between samples of 1 millisecond, and a total displayed time of 0.5 seconds, 512 points is appropriate.

Message About Total Sample Time. To assist in getting the display range and the total sample time correct, the total sample time is computed and shown on the screen after you leave the Sample Rate Option.

Message About Sampling Strategies. To optimize the system performance, the program will select from possible strategies for sampling and displaying the data. The sampling strategy is printed in the sampling parameter block. It is necessary to know whether the data is plotted after each data point is taken or after all data points are taken. The message

"{Real Time Plot}"

indicates that each point will be plotted on the screen as it is taken. The message

"{Delayed Plot }"

indicates that the program will acquire all data first and then display it. This message is updated after you leave the Sample Rate Option. This is not an option you have, but rather a requirement to get the best performance from the hardware system you have. At a rate of 1 each millisecond, your system will take a full set of samples then display the results; i.e. the display should show {Delayed Plot }.

4.4.5 Triggering and Repetition Parameters Block

Mode Option. There are several configurations that can be selected, but for the first part of this example leave the selection as Auto mode. With this choice for the mode, the system will acquire and display results as rapidly as possible requiring no special input to start the sampling process.

4.4.6 Remaining Options

For this exercise no additional selections need to be made. Press the "home" key to return to the command line.

4.5 Sampling and Display

(Move to the command line.) Toggle to the SMPL/DSPLY choice and press the down arrow key.

The command mode and the setup mode display will be cleared and the plot/label image be displayed. Since the mode is auto, the system will start sampling and displaying at its earliest opportunity. When the scan is finished, i.e. the full number of samples taken and results plotted, the system will automatically restart and begin to sample/display the results of a new scan. In the "{Real Time Plot}" mode the subsequent scans erase the old points one by one as the new points are plotted. In the "{Delayed Plot }" mode the plot portion of the display is blanked and rewritten after each new set of data is taken.

4.5.1 Sample/Display Screen

The sample/display screen shows the data and much of the information specified in the setup mode. A hardcopy of the display is shown in Figure 4.4.

Message Line. The upper left corner of the screen is the message line; during sampling, this line indicates the status of the sampling process. It will remind you of the display mode choices (eg real-time plotting vs. delayed plotting) and the triggering selection.

Plot. The data is plotted within the 3 sided box to the right. The axis labels and scales are shown near, but slightly offset, from the "tic" marks on the plot.

Time and Date. The vertical label indicates the source of the signal as well as the chosen label. The time and date are also displayed on the screen. (This is not an accurate clock because it is not continuously updated; depending on the mode of sampling, the time may be updated each sample or only at the beginning and end of a full scan).

Clock Faults. The final piece of information is the log of the faithfulness of the timing of the data. If the system cannot take the data at the proper time it logs a "clock fault". The system does not keep track of how many clock faults there have been, only whether any have occurred. The message ">0 Clock Faults" MUST be interpreted by the user as the data being unreliably timed.

Auto Triggering. In the auto triggering mode the system will continue to sample and display until you interrupt it. Also since the system starts a new scan as soon as it can, there is no synchronization of the start of the scan with the signal. Section 4.8 demonstrates other triggering features. When you have seen enough of the trace interrupt the sampling process as discussed below.

	Sampling Strategy	Triggering Setup	How to Abort
Message While Sampling During Exercise 1	{Delayed Plot }	{Full Scan Trigger:KEYBD}	ESC to Abort

Message While Plotting Data: Scan Completed. Processing Data. Press ESC To Interrupt

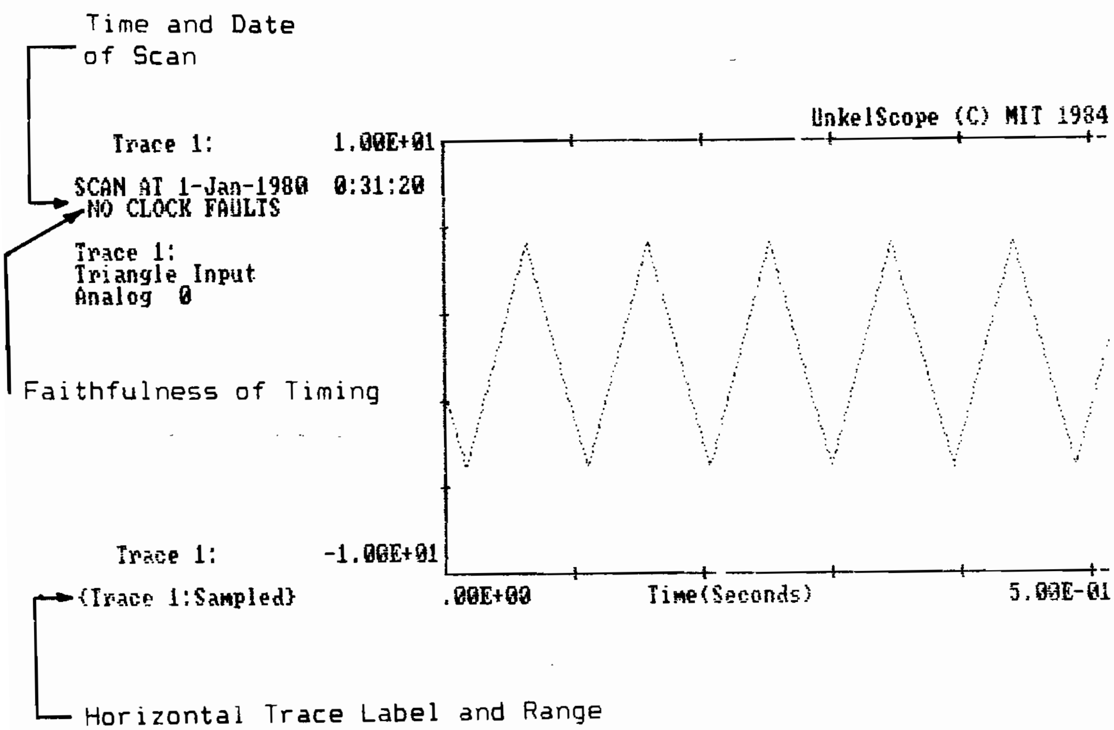


Figure 4.4. Annotated sample/display image of screen as it might appear in Exercise 1.

4.5.2 Stopping the Sampling Process and Returning to the Command Mode

To freeze the display type ESC. You may interrupt either the sampling process or the plotting process. If you have interrupted the sampling process the message line (upper left of screen) will change to read:

Interrupted by Keyboard. Return to Command Mode (Y/N):

Type Y (or y) if you wish to return to the command mode or N to initiate a new scan.

If you have interrupted the plotting process the message line will change to read:

ABORT PLOTTING (Y OR N):

Type Y (or y) if you wish to return to the command mode. If you chose not to abort plotting at this point the system may resume plotting or initiate a new scan depending on the triggering options. If you abort plotting, the system will prompt you again with the message:

Plotting Interrupted. Return To Command Mode (Y/N):

Type Y (or y) to this prompt and you will return to the command mode. If you chose not to return to the command mode at this time the system will initiate a new scan.

4.6. Hard Copy

The full information for the data you have taken is contained in the combination of the sample/display plot and the setup mode display. Before following the instructions in this section make sure the printer is properly set up and at the start of a new page. To obtain copies of these screens on the printer move to the command mode and toggle to the choice PRNT DSPLY, and execute the option (down arrow key). The screen will blank and the display screen will be rewritten with the current data. When the screen is completely written, a "screen dump" to the printer will be made. The time to produce a plot will vary with the particular plotter, and can be as long as a couple of minutes. When the plot is started, the message line will read "Press Y When Plot Done. Type ESC to Abort." Wait until the plot is complete, then press Y. The screen will blank and the command mode lines will be printed. (The setup mode screen will not be written.) To abort the hardcopy plot, type ESC; the printing will abort after printing the next line.

To obtain the hardcopy of the setup screen, toggle to the PRNT SETUP choice and execute the option by pressing the down arrow key. The screen will blank and the setup mode screen will be rewritten and then printed. If your printer output has strange characters in it, you may need to reset the printer properly - see Diagnostics and Fixes section. Type Y when the printer is done; you cannot interrupt the printing of the setup mode screen. You will be left in the command line pointing to PRNT SETUP and the setup mode screen will be displayed. You can now remove the plots from the printer. Make sure to leave the printer "on-line" when you finish.

Printer Problems. If the printer is "offline," out of paper of has some other problem when you attempt to get a hardcopy, the message

```
Write fault error writing device PRN
Abort, Retry, Ignore?
```

will overwrite the part of the screen to the upper left. Set the printer on line and then answer the question with "R" for retry. After the UnkelScope software restarts the plot, you can abort the print if that is appropriate or print the somewhat imperfect image of the screen. Do not respond to the message above with an "A" (or abort) as the operating system will terminate the program as well as the printing (plotting) process; if you do abort the program, you may lose valuable data that has not been stored on the disk.

4.7 Storing and Retrieving Data

The current (last) scan of data taken is now stored only in memory and will be overwritten if you scan again or lost if you quit the program at this point. To store the data for later manipulation and/or replotting for comparison with new data you need to store the data in a disk file. In the command mode, toggle to the SAVE DATA option and execute the option by pressing the down arrow key. You will be prompted for the file name and a 20 character label; the current time and date are automatically stored with the file. You will specify the short file name of the form name.ext where name is up to 6 characters and ext is 3 characters. You should use alphanumeric characters only. The UnkelScope software allows a total of only 10 characters to define the file name. You may wish to use an "ext" of "dat" to avoid confusion of file types. While entering the file name and label, the delete key deletes 1 character, while Ctrl/U (Ctrl and U at same time) deletes the entire entry. The arrow keys cannot be used for "editing" the line. The up arrow, down arrow or return key indicate you have completed an entry. You should write the file name down in case you later wish to recall it. The program makes some checks for valid file names. For example if the file name is blank, the program will put the message "STORE DATA:ERROR IN FILE OPEN OR WRITE" in the message line and place you at the SAVE DATA choice of the command mode.

When the file is successfully completed the cursor will be returned to the SAVE DATA choice and you will be ready to do something else.

The data file can be retrieved later using the program (see Section 4.7.2) but is not directly readable outside the program - for example a "listing" will not be usable. Full access is provided to the data and the techniques are discussed in Section 7. Part 2 and Part 3 of this guide discuss the processing capabilities of UnkelScope.

Directory Listing. You can get a list of the current directory without leaving UnkelScope. In particular, if you request a file that does not exist you will be asked whether you wish to see a directory listing.

Overwriting Existing Files. If you attempt to write over an existing file, a message, in the message line, will request that you to confirm that you wish to destroy the old file and replace it with the new data file.

4.7.1 Store the Data File

Move to the SAVE DATA choice of the command line and press the down arrow key. Type EXMP.DAT to the request for file name. Then type an appropriate label of 20 characters or less. The program will write the data and when done print a message to the message line. The cursor will return to the command line pointing to SAVE DATA.

4.7.2 Retrieve the Data File

The stored traces of a data file are retrieved as "inputs" to Vertical Trace 1 or Vertical Trace 2. The data stored in Trace 1 of the file is selected by moving to the Source line of Vertical Trace 1 and toggling to the choice [Fil Tr 1]. Press the down arrow key and then type the file name in response to the : prompt. The file will be read and the file label printed to the message line. The time and date of when the data file was stored is also printed in the message line.

To display this data, move to the SMPL/DSPLY choice and press the down arrow key. Since no request has been made to take data, the data from the file is displayed until you respond (by typing Y or y) to the question in the message line.

Saving and Retrieving Data for Traces 5,6,7, and 8

If you take data into Traces 5,6,7, or 8 you will need to read the comments below.

Storing Data From Traces 5,6,7, and 8. If data has been sampled into these channels you will be prompted for 2 file names when you Save Data. The first file you store will contain the valid data for Traces 1,2,3, and 4. The second file you store will contain the valid data for Traces 5,6,7, and 8. The message on the screen will indicate which you are currently dealing with.

YOU MUST store the data in Traces 5,6,7 and 8 before you execute any Utilities as the data will be corrupted (storage space is shared) if you do not. If you have taken data into Traces 5,6,7, or 8 and have not yet stored it, you will be warned if you try to enter the Utilities section of the program. You will be given an opportunity to go back and store the data. If you do not store the data before youe enter the Utilities Section data in Traces 5,6,7, and 8 will be destroyed.

Retrieving Data Store in Traces 5,6,7, and 8. The data originally stored in these traces is stored in the "second" file as traces 1,2,3, and 4. For example if you saved this data in a file called "data.pt2", you would retrieve Trace 6 of the originally sampled data by selecting "Fil Tr 2" of file "data.pt2".

4.8 Triggering Options

Triggering allows you to synchronize the UnkelScope to events in the experiment. With the auto triggering you have used so far, there is no control of when the scan begins and stops. This is adequate for some diagnostic purposes but not generally for taking final data. This section will introduce you to some of the additional options relating to triggering. The triggering options control two aspects of the sample and display operation: the manner in which the plot portion is prepared and the synchronization of sampling process to external events.

Triggering Mode.

Auto - The scan starts whenever it can. For the "Delayed Plot" mode, first taken and then plotted, the plot portion of the screen is erased and then the new data plotted. In the "Real Time Display" mode subsequent scans erase points just ahead of the current time. The sampling process is stopped by typing ESC to interrupt the display. There is no synchronization of the scan start to external events.

Normal/Overlay- and Normal/Erase - For these choices a scan is initiated by the "trigger." In both cases the sample/display screen is written and the trigger "armed" when the writing is complete. The scan is started when the trigger (the trigger characteristics are described below) signal is received. After taking a scan, the trigger is reactivated as soon as possible and when a new trigger is received a new scan is initiated. In the normal/overlay mode old data (i.e. previous scans) is not erased from the screen. In the normal/erase mode all data is erased according to the same strategies as in auto mode.

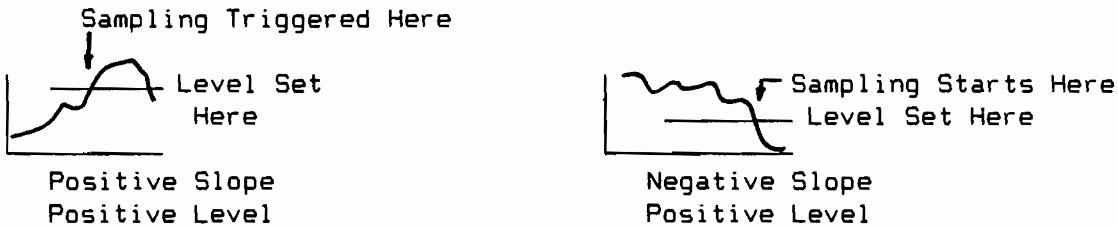
Single Sweep - This choice is similar to the normal/erase mode except that at end of each scan the trigger is not automatically rearmed. Rather, at the end of the scan a message is displayed in the message line and you are given the opportunity to return to the setup mode (to alter options or save data, for example) or to rearm the trigger for a new scan. The message line will read "Return to Command Mode (Y) or Type A to Scan Again:"

Sngl/Trg/Sync - This choice is similar to the single sweep choice except that each trigger pulse causes 1 sample (of each channel if more than one channel) to be taken rather than a full scan of data. The trigger is rearmed as soon as the sample(s) are taken. After all samples are taken, the user is given the choice to restart a new scan or to return to the command mode.

<u>Trigger Characterstics.</u> For modes other than auto, you must select the characteristics of the trigger.

Keyboard - With this choice, the scan is triggered when you type a character. The spacebar is a good choice; just about any character will work, but ESC will abort the scan rather than trigger it.

Analog Channel- This option allows you to select as the trigger one of the analog signals. The triggering channel need not be a sampled channel. In addition to specifying which channel to use, the slope and level at which the trigger is to occur must be specified. The scan is triggered when the signal increases from below the trigger level to above the trigger level. Typing ESC will interrupt the process of waiting for a trigger signal and can be used to abort the process if an inappropriate trigger level has been chosen.



Sketch showing the use of the software trigger.

External - This option allows you to use one of the digital input lines to control the sampling. This option may require special hardware connections and may be better described as a "Gate" rather than a trigger for some hardware options.

4.8.1 Setup Single Sweep, Analog Triggered Scan

Set the source of Vertical Trace 1 as Analog Channel 0.

Now modify the triggering block to synchronize the start of the scan with the signal. In the command mode move to the SETUP MODE choice and execute the option using the up arrow key. Press the arrow key again to move to the mode option in the triggering block.

Mode. Toggle to the single sweep choice and move to the source option line.

Source. Toggle to the Analog 0 source choice and execute the option by pressing the down arrow key.

Slope. Select the + slope choice.

Level. Toggle to a signal level of about 1/2 the maximum signal level. If the triggering input is not also a signal input to one of the traces, the toggling increment and maximum value are chosen as 0.1 and 10 respectively. If the trigger signal source is also chosen for one of the trace inputs, the toggling increment and maximum value will be chosen to be consistent with the values of the span/range settings used in the trace input. After setting the level move to the command line by pressing the home key.

4.8.2 Sample and Display the Synchronized Waveform

Toggle to the SMPL/DSPLY choice and execute the option by pressing the down arrow key. The sample/display screen will be written and the trigger "armed." The sweep will trigger at the appropriate level resulting in a screen image such as shown in Figure 4.5. At the end of the sweep the message line will show

"Return to Command Mode (Y) or Type A To Scan Again:"

Type "A" to rearm the trigger and repeat the process. Note that the scan starts at the same level, unlike when the waveform was scanned using the auto trigger mode.

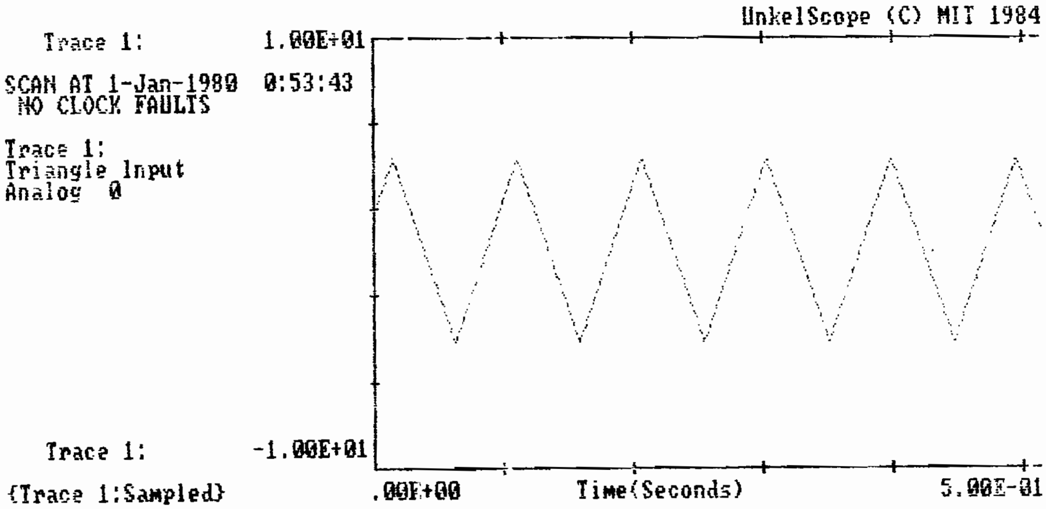
4.8.3 Return to Command Mode

When you are comfortable with the operation, return to the command mode.

4.9 Summary

You have now completed Exercise 1. You have sampled and displayed an analog signal, and obtained hard copy documentation for the data similar to that shown in Figure 4.5. Part of what you have done could have been accomplished with an oscilloscope and camera, or with an x-y plotter or with a stripchart recorder. However, the traces of the data are more explicitly labeled and the data is also stored on the disk for later signal processing.

At this point you can use the system to sample, display and store the data of interest to you. You are encouraged to use the system at this point and to continue with the reading and exercises when you have successfully taken some data.




```
____________ Vertical Trace 1 ____________
Source [Analog  0]  A/D Range [+/- 10]
Label   :Triangle Input
Span    [ 20    v full scale]
Range   [-1.00E+01 to  1.00E+01]

____________ Vertical Trace 2 ____________
Source [None      ]

____________ Horizontal Trace ____________
Source [Time      ]
Label   :Time(Seconds)
Span    [ 0.5 s full scale]
Range   [  .00E+00 to  5.00E-01]

______________ Sampling ______________
Sample Rate [  1 ms        1  khz]
[ 512] Samples  {Scan Time 5.12E-01 s}
{Delayed Plot  } {Processing Inactive}

______ Additional Vertical Traces ______
Tr Input                            A/D
#  Chan  Label:                   Range
3 [none ]
4 [none ]

______________ Triggering ______________
Mode  [Singl Sweep] Source [Analog  0]
Slope  [ + ]         Level  [ 2.00E+00]

______________ Processing ______________
Type [none                    ]
```

Figure 4.5. Documented trace of a 10Hz triangle wave using the software package.

5.0 Computer-Based Sampling Systems

For an analog system measurements are made continuously in time. The resolution of voltage is also continuous although "noise" may cause problems at low signal levels. For the computer-based system, the measurements are made at particular points (or close to points) in time and the resolution in voltage for a given full scale range is limited to a finite level. Further, in the computer system some of the components are shared between the several inputs and this results in timing imperfections. When the computer-based system can sample quickly relative to the change in a signal and the signal level is much larger than the resolution, the computer-based system can be viewed as a direct substitute for the measurement capability of an analog instrument. Many times these conditions are not met so you must understand some of the operating principles of the computer-based system.

This section describes the basic hardware associated with the computer-based system, defining many of the commonly used terms and some of the operational details. The goal is to introduce some of the potential problems in the sampling process. Section 6.0 contains Exercise 2 which demonstrates some of these issues using the UnkelScope software. Exercise 2 also introduces additional features of the software, including the x-y or cross-plotting capabilities.

5.1 Terminology

Sample. One value of an analog signal.

Sweep. A set of values of all sampled signals. Values taken as close to the same time as possible.

Scan. A set of sweeps separated by a specified interval.

5.2 Signal Flow, Timing and Overall Control

This section describes the operation of a multiple input analog-to-digital system. There are several variations of A/D systems and where possible an indication of other configurations is given. Also realize that the details of performance also depend on the hardware and software implementation. The configuration considered here is shown schematically in Figure 5.1.

5.2.1 Signal Flow

The analog signals are input to one side of the solid state switch box called a multiplexer or MUX for short. The switches in the MUX are controlled by the system hardware to connect the output to a selected analog input signal.

The signal output of the MUX is connected to a sample and hold (S/H) amplifier; this device has two "states." In the sample state the output follows the input as closely as possible. When switched from the sample state to the hold state the device freezes the output to the current value. As with the MUX the state of the S/H is controlled by the system hardware.

The output of the S/H is connected to the analog-to-digital (A/D) converter, perhaps with an amplifier in between. The A/D converter is the device that converts the analog signal into a digital number. The signal is resolved into a finite number of intervals, each representing a finite voltage range. For example if 4096 intervals are allowed in the voltage range from 0 to 10 volts, the device will be able to resolve 2.442 millivolts and cannot tell the difference between 4.8 millivolts and 2.5 millivolts both being determined as 2.442 millivolts. In essence the A/D board truncates the data. This resolving power is referred to as the least-significant-bit interval. The device may also have offsets or errors in gain resulting from the analog circuitry.

The output of the A/D is a number and this number is stored in local memory called a register. This register is accessible to the computer so that values from it can be moved to the computer for long term storage.

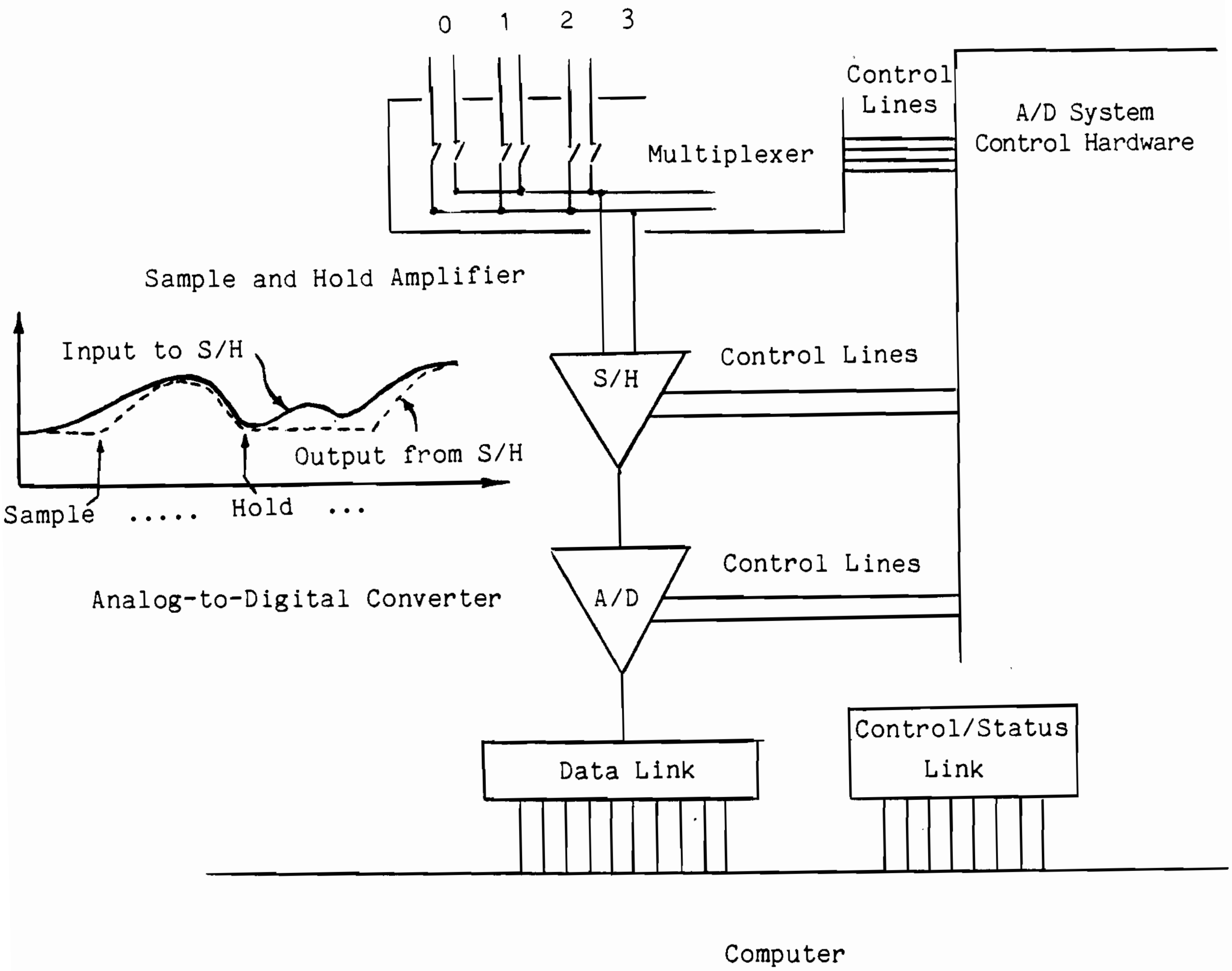


Figure 5.1 Schematic representation of analog-to-digital converter system showing the signal flow.

5.2.2 Timing

The data acquisition board has the appropriate hardware circuits to control the timing of the MUX, S/H, A/D and register circuits once the commands to perform a conversion on a specified channel are received. The details of this design are beyond the scope of this guide. While the user can leave the inner timing issue to the designer, the rate of data sampling, or the delay time between samples needs to be set by the user. For this purpose, the data acquisition system has a programmable clock that can generate a signal at prescribed intervals. The computer or the A/D system hardware uses this timing signal to coordinate the overall operation of taking data.

5.2.3 Overall Coordination

There are several ways to coordinate the data taking process, especially in the case of multiple channel sampling. The method described here is one used in the software implementation. The process is divided into two parts: a setup portion and a running portion. The sequence, in semi-flow chart notation, is shown in Figure 5.2. The setup consists of assembling the information about the delay time between samples and the multiplexer inputs to be sampled at the end of each interval in a manner understood by the computer and A/D system.

The clock is first set-up, but not necessarily started. If specified, the system waits for a trigger and the starts the clock. If no trigger is specified, the clock starts immediately. The next step is to wait for the first clock "tic" i.e., to wait until the Time Between Samples has elapsed. At this time the analog signals are "acquired" in a sequence as quickly as possible. Acquiring an analog signal consists of telling the system which channel and waiting until the device has completed its task. This process is controlled by the hardware of the A/D system. When the conversion is done the software moves the result from the register to a more permanent storage place, usually a memory location. When all channels of a sweep are sampled the software system may perform other tasks such as plotting the results if there is time. But the system must give control back to the A/D part of the operation before the next clock tic or the timing of the process will not be faithful. The whole process continues until the required number of samples is taken.

A. SETUP PROCESS
   1) For each sample in a sweep
      Gather together in the proper format the channel #, etc.
   2) Convert the desired time between samples to the proper format.
   3) Establish scan or sweep triggering strategy

B. SAMPLING PROCESS

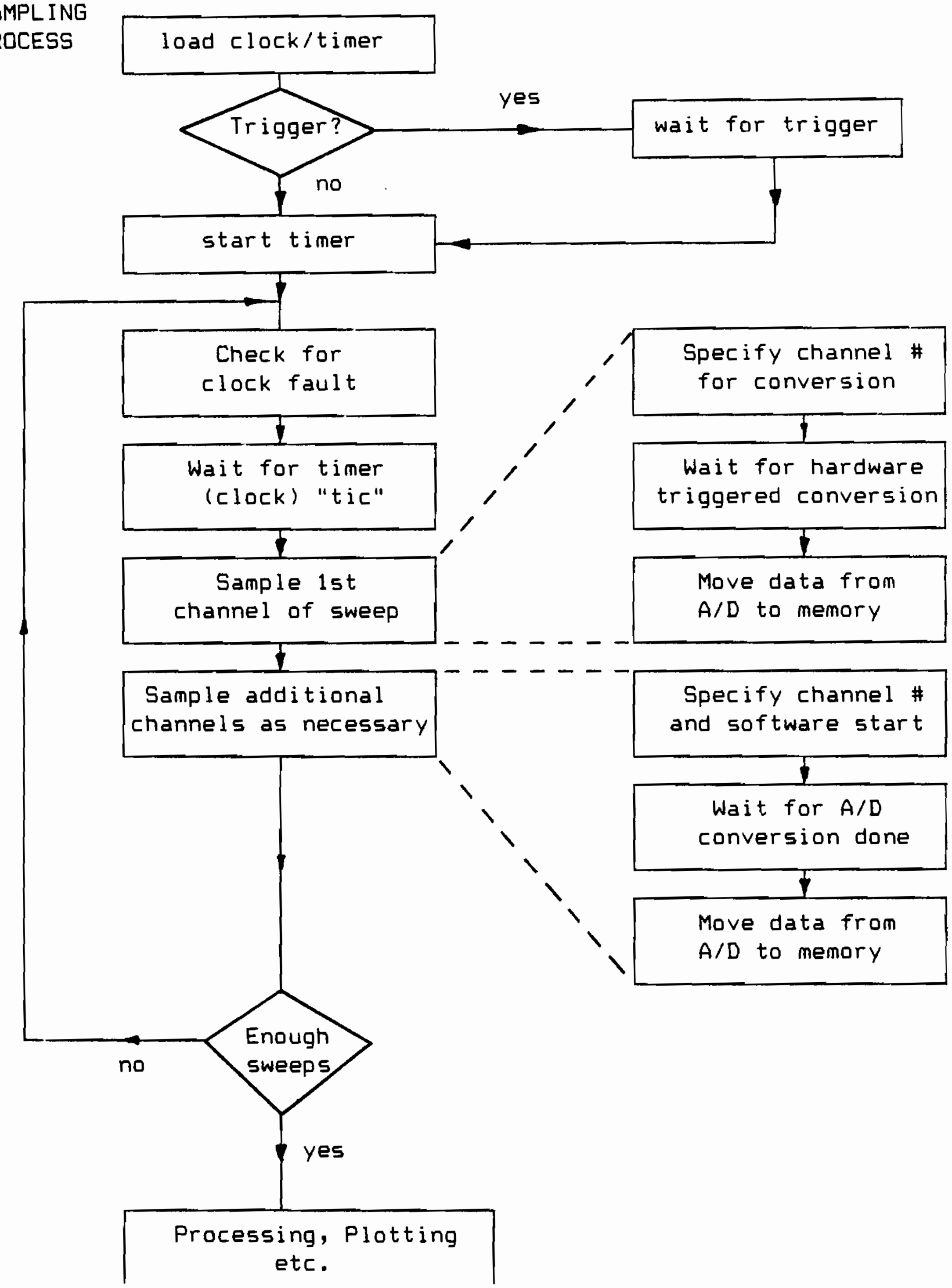


Figure 5.2 Semi-flow chart representation of software control of data sampling process.

5.3 Timing Issues

The "clock fault" in the UnkelScope software indicates a failure to time the sampling process. However, this is only a check that everything gets done within the interval, (i.e., before the second tic) and does not ensure any particular timing of the events within the interval. The timing is shown roughly in Figure 5.3. The clock is used to start a "sweep" rather than to start each conversion; this keeps the samples as closely spaced as possible and concentrates the A/D sampling activities. Nonetheless, as will be shown in Exercise 2, there is a delay between each sample in one sweep and there can be significant time "jitter" when the system is used to sample close to its maximum speed. Note that if the commands to the system were precisely timed, the actual time at which the signal was taken would still show some jitter. A better A/D system, but a more costly one too, is to use a sample and hold amplifier in front of each analog signal and to switch all these to hold at the same time. The delays and jitter of the sampling process would then not result in errors when the signal was acquired; some jitter would still be present in the hardware, but this would be small in comparison with that in the system without simultaneous sample and hold. In most cases the extra cost of a simultaneous sample and hold is not justified, but in some cases it becomes a necessity.

5.4 A/D Resolution Issues

As shown in Figure 5.4, the A/D converter compares the voltage to be digitized to a sequence of known voltages and using a voltage comparator makes the binary decision of whether the input analog signal is larger or smaller than the known voltage. Most moderate speed (10 mirco-sec) A/D converters use the "successive approximation" technique which locates the signal within a range which decreases in size by a factor of 2 each step of the approximation. The first step determines if the signal is more or less than half the full scale. The second step isolates which 1/4 section the signal is in. The process continues n times, with each step requiring an additional time interval and an additional "bit" of storage in the A/D register. When the process is complete the device signals that it is done and has the "n bit" number in the register for the computer to read.

For a chosen full scale range of -10 volts to +10 volts, the distance between the smallest steps is about 4.88 millivolts. At this full scale range the device is unable to distinguish between voltages separated by less than this interval.

5.5 Summary

The computer system measures the analog signals at specific times and with a resolution limited by the number of "bits" of information stored about the signal. Too small a signal will not register a difference in the A/D; too large a signal will saturate the A/D. For the software/hardware combination described here, faithful timing of the data is ensured only in the sense that all measurements are taken within the desired interval and no precise timing of events within the interval is ensured. For a 12-bit converter, the resolution is at best 1 part in 4096 of the full scale voltage range. Exercise 2 described in Section 6 demonstrates these issues as reflected in the present hardware/software combination.

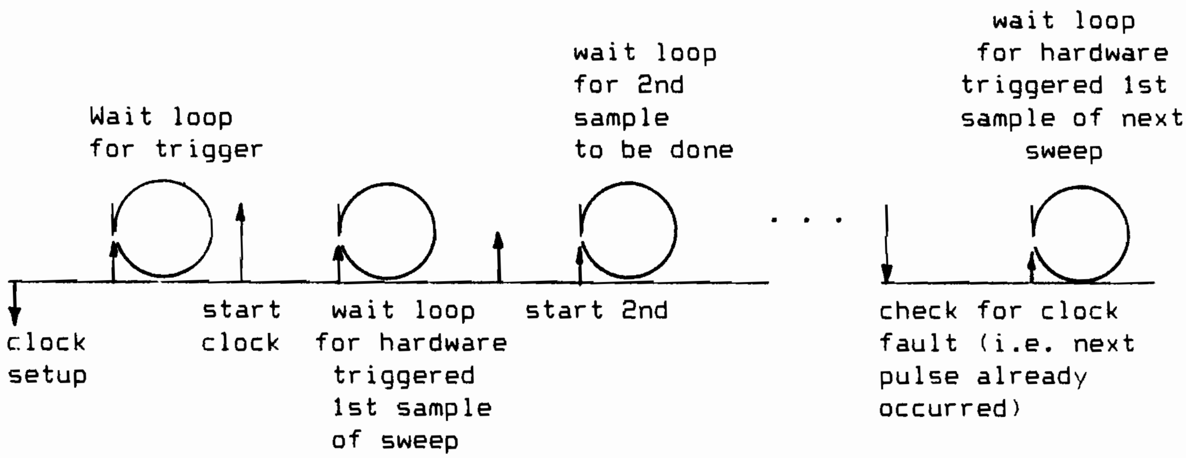


Figure 5.3 Schematic of timing of data acquisition process.

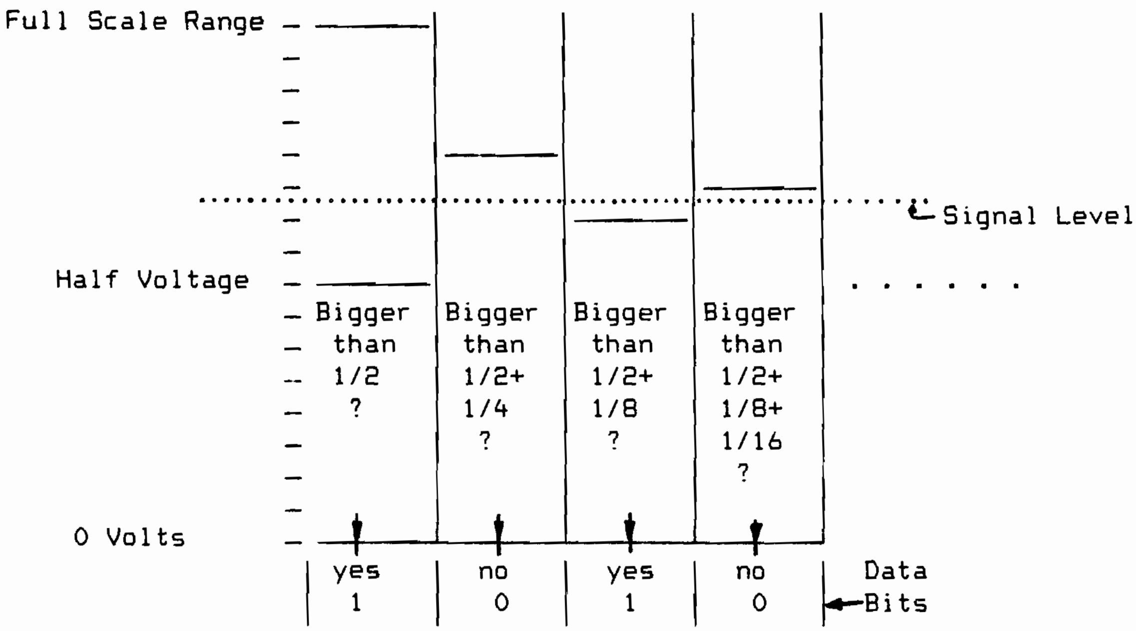


Figure 5.4 Representation of successive analog-to-digital converter.

6.0 Exercise 2 - Issues of Discrete Sampling Systems and Use Cross Plot and Save/Get Setup Features

This exercise demonstrates the limitations - in sampling speed and signal resolution - of the A/D system so you can use the system properly. Do not interpret the comments or "problems" as defects in your equipment. Knowing the limitations of any device is an important part of being able to use it properly. You will also learn to use the cross-plotting (x vs y) capability and to save and restore the parameters of the setup screen.

This exercise assumes that Exercise 1 has been performed and that Sections 1-5 of the guide have been read.

Additional Equipment. In addition to the computer based system you will need a source of time varying analog voltage (frequency from 10 to 1,000 Hz, level of about 2.5 volts peak to peak with triangle or sinusoidal waveform). You will need an appropriate connector from the source and the analog-to-digital input of the computer. In addition you will need a connector to which a 1 mega-ohm resistor is attached across the signal leads.

6.1 Setup and Checkout of Equipment

If you are running the program from a floppy disk system, you should have a formatted disk in the "B" drive.

Hook up the analog voltage source to Channel 0 of the A/D system, set the input to triangle wave, about 5 volt peak-to-peak and no DC offset (or similar settings as appropriate). Enter UnkelScope and make sure that the equipment is operating properly.

6.2 Saving and Getting Setup Parameters

Often an experiment requires that several configurations of the sampling parameters be used at different times. It is also useful to be able to make a quick check on a signal source and see that it is operating properly. Finally it is convenient to be able to restore the settings after they have been altered by some other user or "lost" when the system was powered down. The SAVE SETUP and GET SETUP features allow you to quickly restore settings by first saving them in a disk file and later getting them by reading the disk file.

In this Section you will save a setup files and use several setup files stored previously to complete the exercises quickly and efficiently.

6.2.1 A Checkout File

In a shared system it is often useful to make a quick check that nothing has been disturbed or has broken or that things have been properly restored. This is what you should have done as part of completing Section 6.1. To make this easier, save the settings you used in a file with the name CHK.SET. Then each time you power up you can easily check the operation of the system without having to reset the parameters of the Setup Screen one by one.

Save_Setup. In the command mode, toggle to the SAVE SETUP choice and execute the option by pressing the down arrow key. The system will ask for a file name by printing in the message line

Save Setup.File Name(10 char):

Since the line ends with a :, an alphanumeric response is expected. Use the delete key and Ctrl/U to delete one character or the whole entry respectively. Use the down arrow key (or return) to indicate theat the entry is complete. The system will write a file (as with the data file, the file is directly readable only through the program). When this done UnkelScope will respond with

Save Setup.File Written

and move you back to the SAVE SETUP choice of the command option. If you execute the save setup option by mistake, a blank file name will return you to the SAVE SETUP choice without saving anything. An error message will appear in the message line; ignore the error and proceed to do what you wished to do.

Get_Setup. Move to the GET SETUP choice and press the down arrow key. The system will print a request in the message line:

Get Setup.File Name(10 char):

Enter the name CHK.SET. For a file name lowercase characters are converted to uppercase. If you type the name of a file and it does not exist, the program will ask you if you wish to see a listing of the directory. If you respond "y" it will display the contents of the directory in the lower half of the screen.

After entering the file name, the file will be read and the Setup Screen rewritten with the settings restored from the file. The settings present when GET SETUP is called will be lost (unless they were previously stored in a disk file).

6.2.2 Setup Files for Exercise 2

This exercise assumes that you have the signal source set at 5 volts peak-to-peak, triangle wave and 10 Hz and that you can select frequencies of 1 Hz, 10 Hz, 100 Hz, 1 kHz and 10 kHz. If this is not the case, some modifications to the setup files may be necessary. There are 5 setup files saved on the disk to be used in this exercise. They are

EX2SE1.SET - digitization error
EX2SE2.SET - name check
EX2SE3.SET - cross plot
EX2SE4.SET - clock faults
EX2SE5.SET - aliasing issues

Note that these files are on the "program" diskette or in the subdirectory "c:\uscope".

6.3 Digitization Error, Timing Errors and Aliasing

The 3 parts of this exercise display the 3 most common "problems" associated with computer based sampling. The first two, timing and digitization errors have already been discussed in general terms in Section 5 and are covered in Exercises 2a and 2b. Aliasing is a misinterpretation of the sampled data that results from sampling at too slow a rate relative to the time variation of the input signal. Exercise 2c demonstrates this problem.

6.3.1 Exercise 2a: Digitization Error

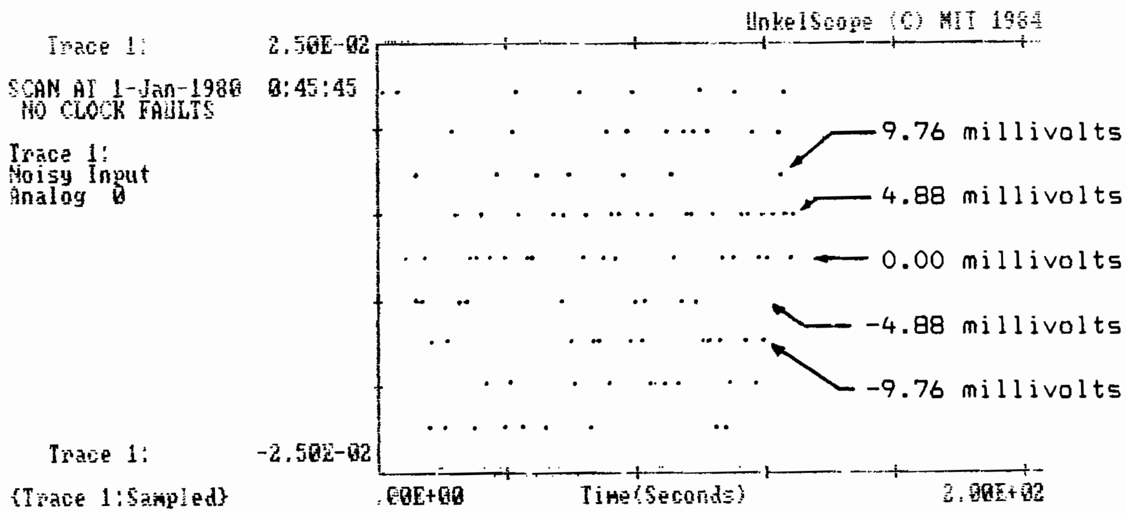
In addition to analog issues, i.e., offsets, gain errors and drifts in the A/D system and transducer system, the A/D system has a limitation on the accuracy imposed by the number of "bits" or divisions the interval is divided into.

Analog Input. Connect to Analog Input 0 a long cable (at least 2 ft.) with a 1 mega-ohm resistor across one end. This will be a source of low voltage "noise." Place the cable near the video monitor to make sure you will get some noise.

Retrieve and Examine EX2SE1.SET. In the command mode, toggle to the GET SETUP option and press the down arrow key. Enter the file name EX2SE1.SET. The system will read the file and write the settings to the screen. This setup selects Analog Input 0 as input to Vertical Trace 1 and selects a fine range (50 mV) centered at zero. If you select the A/D Range you should select the largest range (poorest resolution.) A delay between samples of 1 second is chosen and at this rate the sample/display screen will show the digital value of each sample taken. Triggering is set to single sweep, keyboard initiated. 128 samples are taken with a total time displayed of 200 seconds.

Sample and Evaluate. Toggle to the SMPL/DSPLY option and press the down arrow key. The system will sample and display the points on the screen. Below the label for Trace 1 on the Sample/Display screen will be the number of the sample just taken and the value of the last sample of the input to Vertical Trace Input. Thus when sampling slowly (<1 sec delay) the system also functions as a digital voltmeter.

Figure 6.1 shows the results typical of this exercise. Note on the figure and your display that only discrete levels of voltage can be observed and that these levels are roughly 4.883 millivolts for a system with +/- 10 voltage range.



* Note: When sampling at rates slower than once per second, each sampled voltage level is shown numerically in the region below the vertical trace label.

```
---------- Vertical Trace 1 ----------       ------ Additional Vertical Traces ------
Source [Analog  0]  A/D Range [+/- 10]       Tr Input                            A/D
Label  :Noisy Input                          #  Chan  Label:                   Range
Span   [50   mv full scale]                  3 [none ]
Range  [-2.50E-02 to  2.50E-02]              4 [none ]

---------- Vertical Trace 2 ----------
Source [None      ]

                                             ------------ Triggering ------------
                                             Mode  [Singl Sweep] Source [Keyboard ]

---------- Horizontal Trace ----------       ------------ Processing ------------
Source [Time      ]                          Type [none                    ]
Label  :Time(Seconds)
Span   [200  s full scale]
Range  [  .00E+00 to  2.00E+02]

-------------- Sampling --------------
Sample Rate [  1 s          1 hz]
[ 128] Samples  {Scan Time 1.28E+02 s}
{Real Time Plot} {Processing Active  }
```

Figure 6.1 Display and setup screen results typical of Exercise 2a. Note the discrete voltage levels, separated by about 4.88 millivolts for the full scale range chosen here.

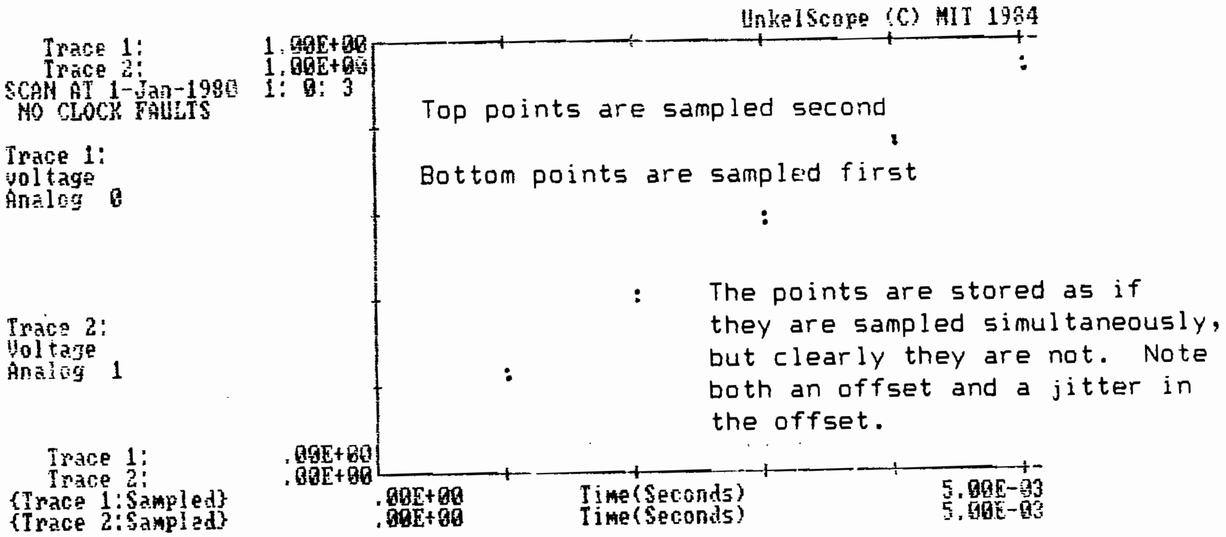
6.3.2 Exercise 2b: Timing Errors

Section 5 discussed the timing of data samples; this exercise shows these issues in some detail by sampling the same signal on two analog inputs.

Analog Input. Connect the function generator output signal to the inputs of both Analog Channel 0 and Channel 1. Set the function generator to 5 volts full scale, triangle wave and 10 Hz or similar settings appropriate to your configuration.

Retrieve and Examine EX2SE2. Toggle to GET SETUP, press down arrow key, enter the name EX2SE2.SET and press the down arrow key. This setup selects Analog Channel 1 in Trace 1, with a range of -5 to +5 volts and Analog Channel 2 in Trace 2 with the same range. Single sweep with Analog Channel 1 as the trigger source is selected; triggering is on positive slope and at a level of 0.0 volts. The delay between samples is 1.0 millisecond (1 k Hz rate) and the display is set to show only the first 5 milliseconds of data.

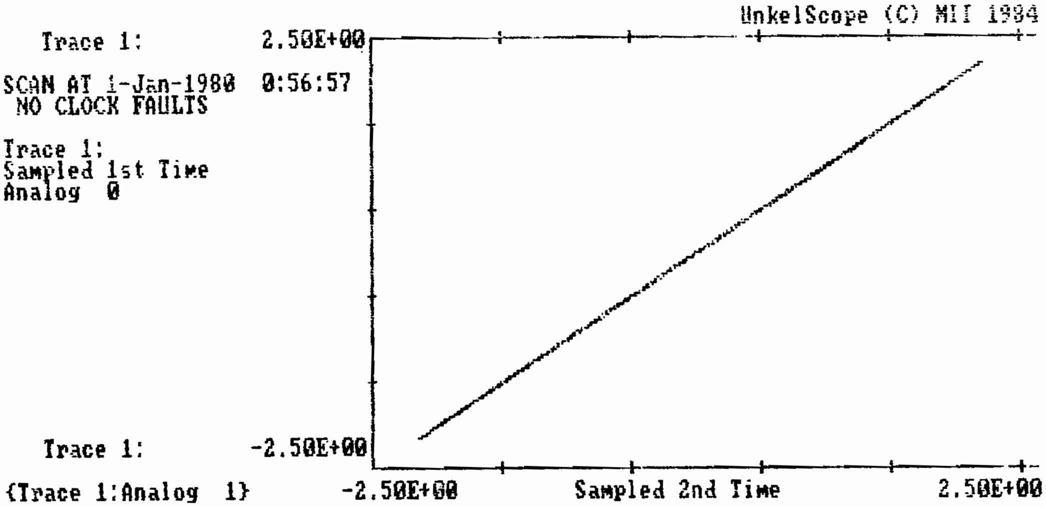
Sample and Display 10 Hz Triangle Wave. Toggle to SMPL/DSPLY and press the down arrow key. The system will begin to sample, taking all the data and then plotting all points from Trace 1 then all the points from Trace 2. If the two samples were taken simultaneously, as assumed in the way the points are plotted and the data stored, the two plots would be be directly on top of one another. You will need to watch carefully to see if this is the case. Repeat several times if necessary. Figure 6.2 shows a result typical for this situation, and indicates: a) that the samples are not taken simultaneously; and, b) that the timing varies slightly from sweep to sweep within the scan.




```
Vertical Trace 1
Source [Analog  0]  A/D Range [+/- 10]
Label   :Voltage
Span    [  1    v full scale]
Range   [  .00E+00 to  1.00E+00]

Additional Vertical Traces
Tr Input                        A/D
#  Chan  Label:                 Range
3 [none ]
4 [none ]

Vertical Trace 2
Source [Analog  1]  A/D Range [+/- 10]
Label   :Voltage
Span    [  1    v full scale]
Range   [  .00E+00 to  1.00E+00]

Triggering
Mode  [Singl Sweep] Source [Analog  0]
Slope  [ + ]        Level  [  .00E+00]

Horizontal Trace
Source [Time      ]
Label   :Time(Seconds)
Span    [ 5  ms full scale]
Range   [  .00E+00 to  5.00E-03]

Processing
Type [none                      ]

Sampling
Sample Rate [  1 ms        1  khz]
[1024] Samples  {Scan Time 1.02E+00 s}
{Delayed Plot  } {Processing Inactive}
```

Figure 6.2. Typical results for the first part of Exercise 2b. Note that the values sampled by the two traces are not taken at exactly the same time.

Cross Plotting to Determe the Sampling Speed. A more graphic way to present these results is to plot the first sampling of the signal vs. the second sampling of the signal. This is done by choosing an analog input rather than time to the Horizontal Trace 1 Source. Get the setup file EX2SE3.SET. Vertical Trace 1 is set for Analog 0 with a range of -5 to +5 volts. No signals are put in Traces 2, 3 or 4. The input to the Horizontal Trace 1 is Analog 1. The sampling rate is 1 millisecond (1 kHz). Toggle to SMPL/DSPLY and press the down arrow key. Set the function generator to a frequency of 10 Hz with a sinusoidal output of 5 volts peak to peak. Trigger a scan by pressing a character (spacebar for example) on the keyboard. Figure 6.3 shows a typical result with all the data lying on a straight line with unity slope. This suggests that the two samples of the same signal have been taken closely spaced relative to the changes in the signal. This agrees with the earlier observations.



Vertical Trace 1
Source [Analog 0] A/D Range [+/- 10]
Label :Sampled 1st Time
Span [5 v full scale]
Range [-2.50E+00 to 2.50E+00]

Additional Vertical Traces
Tr Input A/D
Chan Label: Range
3 [none]
4 [none]

Vertical Trace 2
Source [None]

Triggering
Mode [Singl Sweep] Source [Keyboard]

Horizontal Trace
Source [Analog 1] A/D Range [+/- 10]
Label :Sampled 2nd Time
Span [5 v full scale]
Range [-2.50E+00 to 2.50E+00]

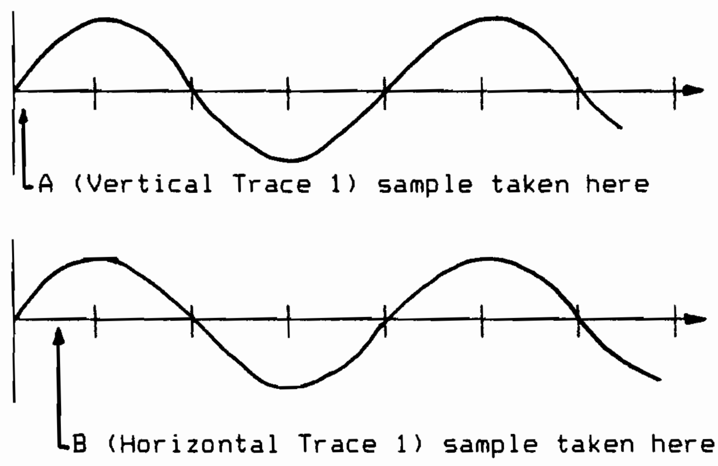
Processing
Type [none]

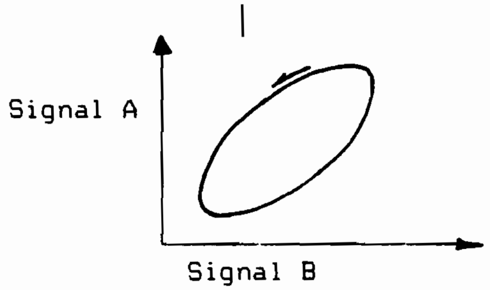
Sampling
Sample Rate [1 ms 1 khz]
[1024] Samples (Scan Time 1.02E+00 s)
(Delayed Plot) (Processing Inactive)

Figure 6.3 Result of cross plotting the two samples of the same input.

Consider the sampling of two signals which happen to be the same signal. A cross plot of two signals is termed a lissajous figure and can be used to contrast the frequency and phase of a pair of signals. For perfect signal acquisition a cross-plot of a signal versus itself will be a unity slope straight line. With the exception of systems with simultaneous sample and hold, the two samples are not taken simultaneously. Further, at best only an estimate of the time at which samples are taken can be made. The present software plots and stores the points as if they were made simultaneously - the result is that the cross plot displays a phase difference between the two sets of samples of the same signal. The situation is indicated schematically in the sketch below.

Take a scan or scans with the function generator at frequencies of 100 Hz, 1000 Hz and then 2000 Hz. At 100 Hz the result shows a slight elliptical shape indicating the non-simultaneity of the two samplings of the same signal. Note also that some points do not lie on the ellipse; this is caused by the jitter in the timing of the two samples. Note that as the frequency increases the ellipse becomes more circular. If you adjust the frequency of the input to get a circle, the effective delay between samples will be t(B to A) $\sim 1/4f_{effective}$. This is one way to check out the effective conversion speed of the system.





Apparent phase shift because 2nd sample is NOT taken at exactly the same time.

Note that a circle would indicate a 90 degree phase difference.

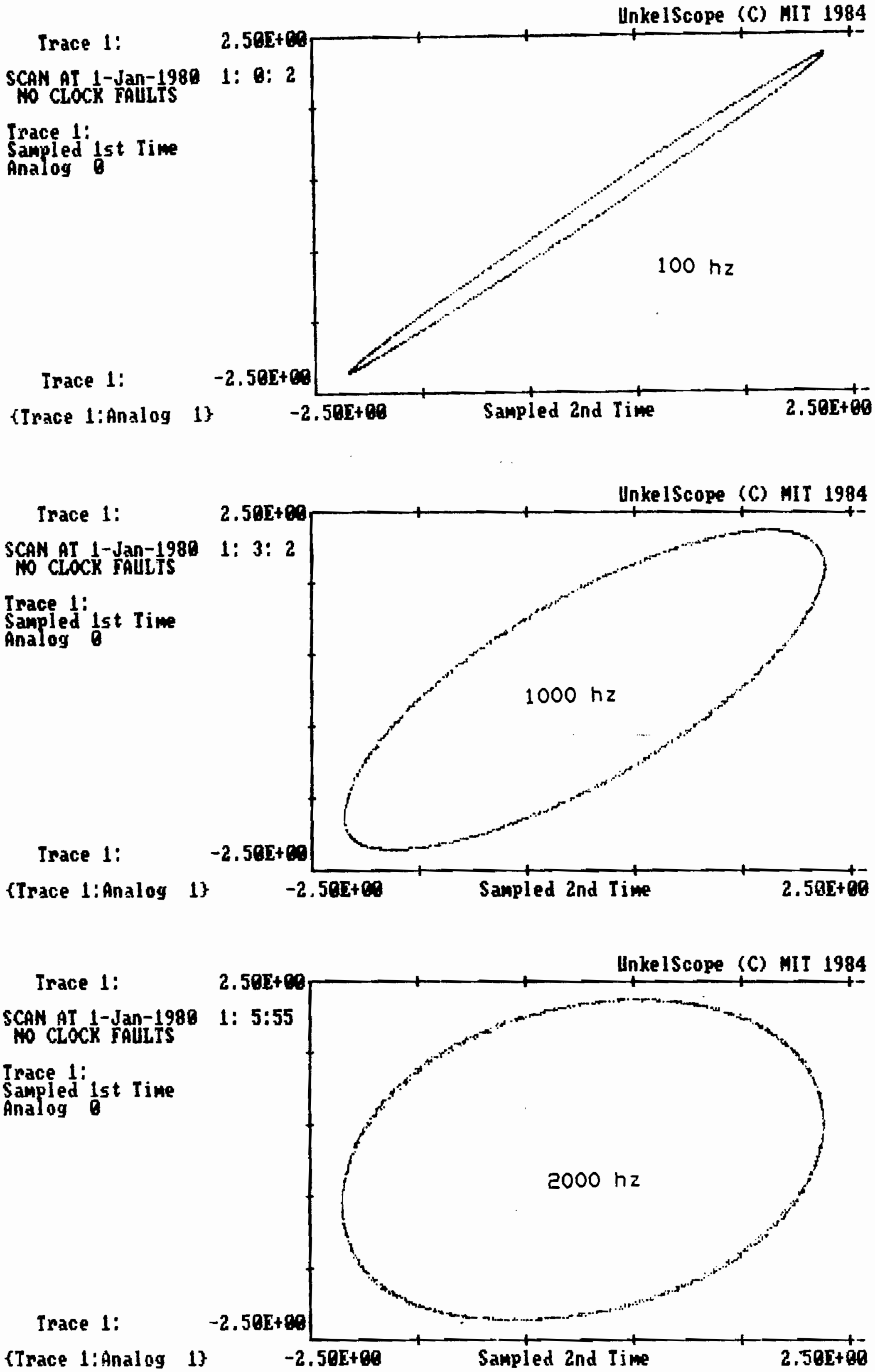
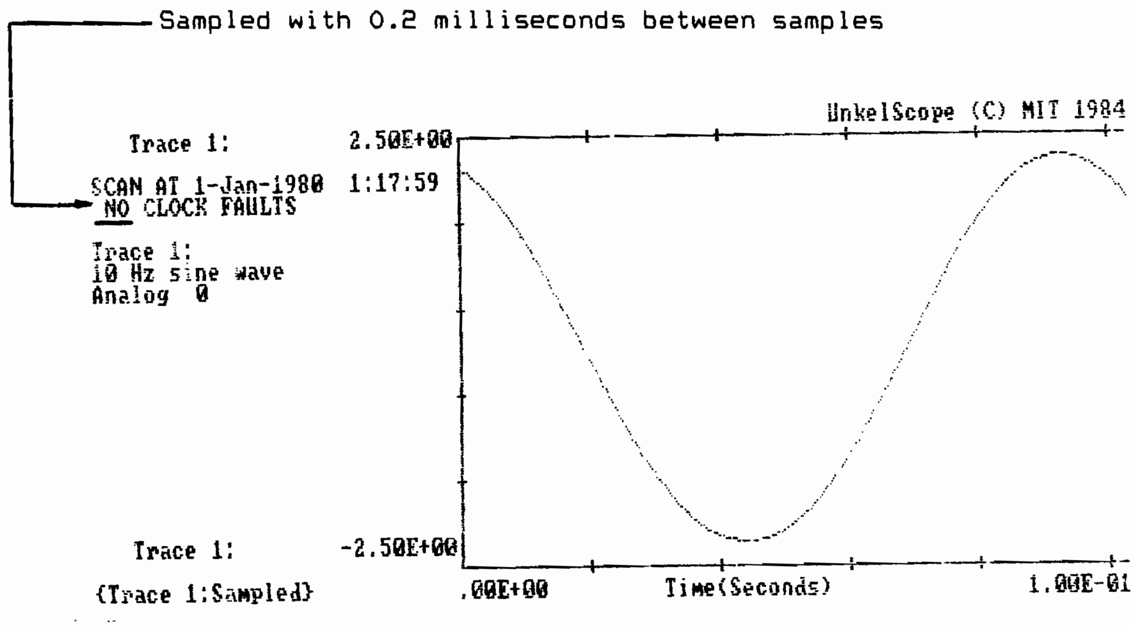


Figure 6.4 Typical cross plots obtained for input frequencies of 100 Hz, 1000 Hz and 2000 Hz.

Clock_Faults. The previous results have demonstrated the errors in timing within a sweep of data; while these can be better estimated and reduced by software/hardware changes they cannot be completely removed. A more gross timing fault is when the sweep cannot be completed before the next one should start. This type of error is detected by the present software/hardware combination. To show the seriousness of the problem set the function generator to give a 10 Hz triangle wave of reasonable amplitude. Get the setup file EX2SE4.SET which is set to sample Analog Channel 0 with a delay 0.5 ms and display 0.2 sec of data, that is 1 cycle of the input. Sample the data and note that the frequency of the signal is correct and that no clock faults are recorded; the trace should be as in Figure 6.5.

Return to the setup mode, decrease the time delay and observe the signal again. Do this until the system reports clock faults and the trace will change qualitatively. When this happens you must view the timing of the data as incorrect. The voltage values may be values at some time, but you cannot tell when.



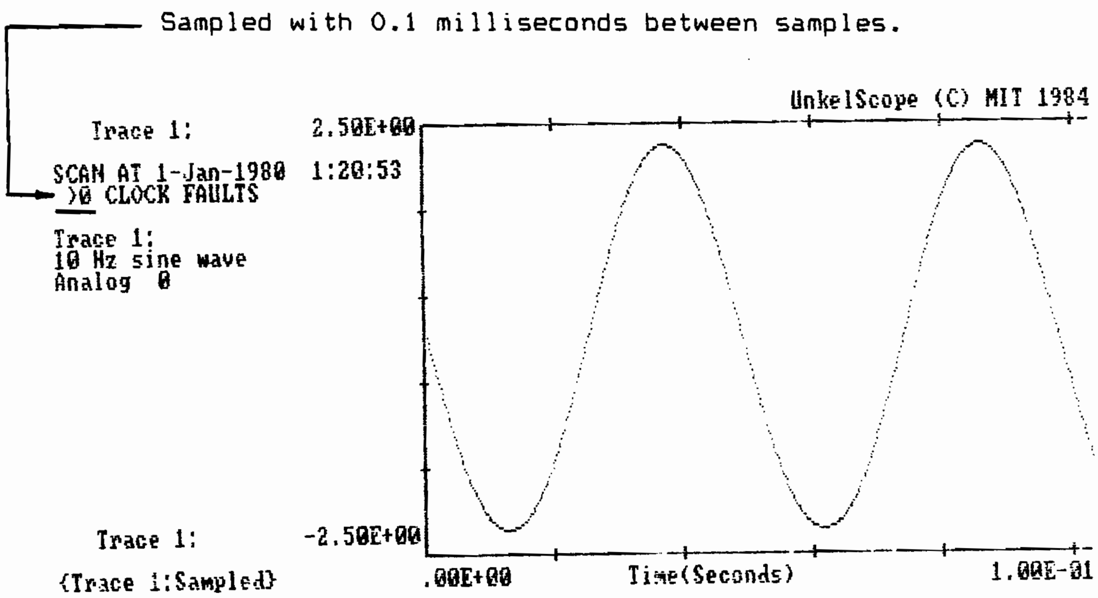


Figure 6.5 Plots showing the meaning and seriousness of clock faults. The input is the same for each case, but the trace showing clock faults shows an incorrect result.

6.3.3 Exercise 2c: Aliasing

The previous section demonstrated some of the timing problems that arise whenever the system needs to scan at close to its maximum rate because the signals are rapidly changing. The problem of aliasing - misinterpretation of a high frequency fluctuation as a low frequency fluctuation - is another problem associated with discrete time sampling. Aliasing results when the rate of sampling is too slow relative to a significant frequency of fluctuation of the signal. Unfortunately, it is not simply a case of "losing" the high frequency information, but rather a misinterpretation of the response.

This part of the exercise demonstrates the problem of aliasing.

Set_Function_Generator_Frequency. Use UnkelScope to obtain a sinusoidal waveform with frequency of 10 Hz. Since the dial on the function generator may not be accurate you should use the program to allow you to adjust the frequency until it is close to 10 Hz. If you wish you can use the setup file EX2SE5.SET.

Sample_At_A_Rate_of_0.5_milliseconds_(2k_Hz). Adjust the time display range to 0.5 seconds and take 1024 points with a time between samples of 0.5 millisecond (2 kHz). You should see approximately 5 repetitions of the sinewave.

Sample_At_A_Rate_of_0.1_seconds_(10_Hz). Now, adjust the time display range to 100 seconds and the time between samples of 0.1 seconds. Sample the 10 Hz sinewave and observe the output. If the period of the input sinewave were to match exactly the time between samples, the apparent result would be a constant voltage value. With the two signals very close in frequency the apparent signal is a sinewave with a very low frequency - a period of $1/(f_{input}-f_{sampling})$. For the results shown in Figure 6.6 the input frequency was tuned to be very close to the sampling frequency and the apparent frequency of the input is about 0.01 Hz.

Sampling at twice the frequency of the input (time between samples of 0.05 seconds) results in the last trace of Figure 6.6. In this case the just inadequate sampling rate is indicated by the "double sinewave".

Aliasing can be prevented by choosing a sample speed at least 2 times the maximum frequency of the source, or by providing a filter that removes components of the signal above 1/2 the sampling frequency. In any case it is useful to sample the input signal at a high sampling rate to determine if aliasing will be a problem at the sampling rate you wish to use. If you intend to sample at the highest speed of your computer system, you may need to measure the response at higher frequencies with some other device.

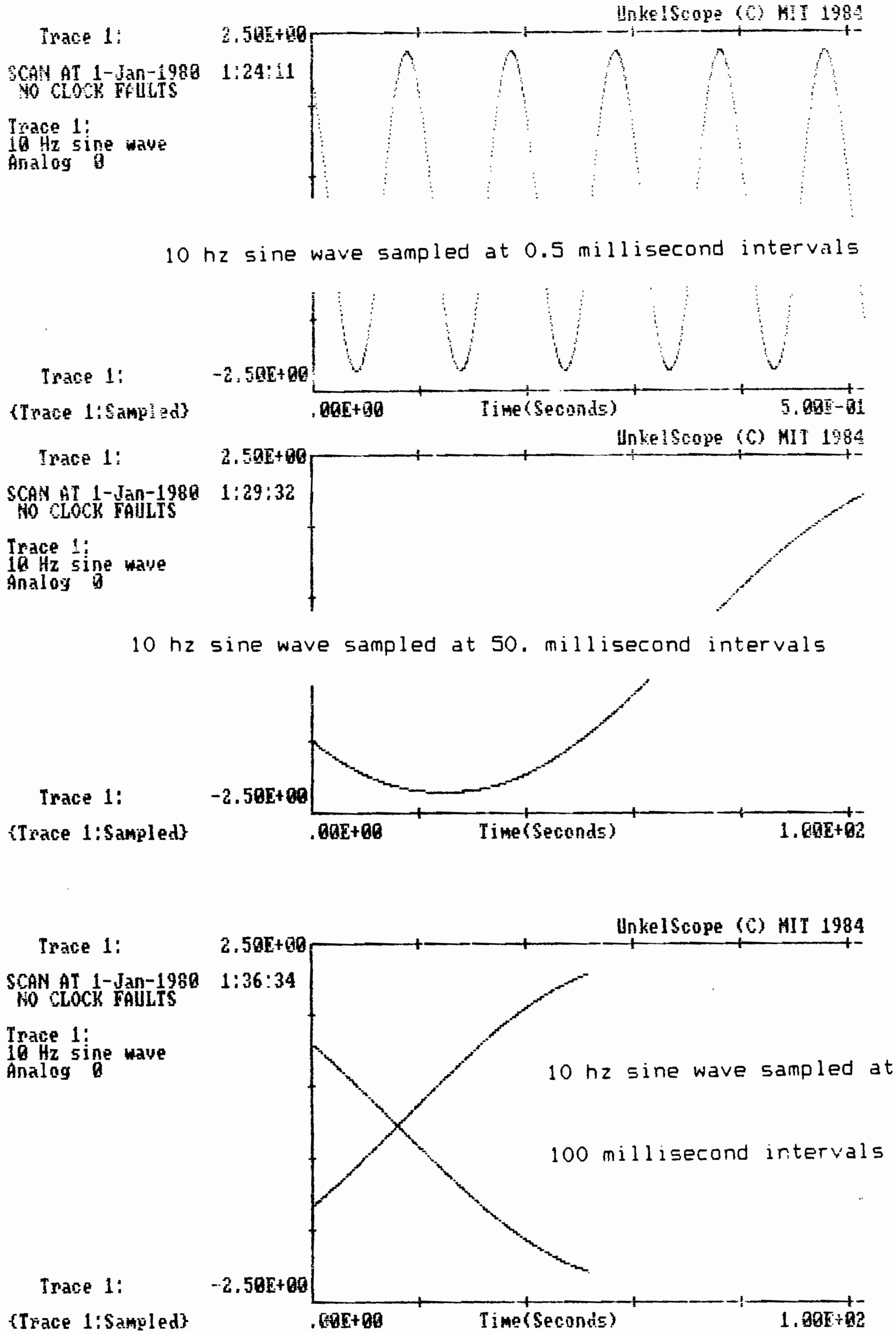


Figure 6.6 Results obtained in aliasing exercise. When sampling too slowly, high frequency signals are aliased and look like low frequency signals. The input for all results shown above is the same 10 Hz sinewave.

6.4 Summary

In this section, four potential "problems" of computer-based sampling have been demonstrated using the computer system. The ultimate resolution of the device has been shown to be a discrete "least significant bit" value. Two types of timing problems have been shown. The first timing related problem was that when several channels are sampled during each sampling interval the samples are not exactly timed and are not simultaneous, at least for the software/hardware combination discussed here (excepting simultaneous sample and hold boards.) The second type of timing problem was whether all the samples could be taken within the specified interval. The UnkelScope software detects faults of this type and logs them as clock faults. When clock faults are logged the timing of the data is unfaithful. Aliasing was the final "problem" displayed and can be avoided by sampling at least twice the frequency of the highest component in the signal. Now you know the potential problems so you can avoid them and use your hardware/software to its correctly.

Also you learned to use the Get/Save setup feature to reduce the effort in switching from one configuration to another.

7.0 Offline Manipulation of UnkelScope Data Files

The UnkelScope Level 1 software allows you to easily sample, display and store data, but does not provide data processing capabilities. UnkelScope Level 2 provides a variety of data processing facilities and can serve the majority of your needs. However, even with extensive data processing capabilities provided, it is important to provide complete access to the data and to make it easy to transport the data to other programs and to other computers. The data files stored by UnkelScope are "unformatted" or "binary" files and cannot be examined directly by an editor, or listed to the screen or printed on the printer. In addition, they can be transferred directly to another computer only in special cases. The "binary" format is used because files take up considerably less space and are read and written much more quickly than a corresponding "ascii" or "formatted" file which is directly readable.

This section describes how to make use of the data from UnkelScope outside the program itself. If your interest is in data processing and you have the Level 2 version of the software you should read Section 8 to see if the existing software will accomplish what you want. If you have the Level 1 version or you do not choose to use the Level 2 software you should be able to accomplish your goals using the techniques in this Section. But, you will need to use/learn other programs or programming languages.

Offline processing programs, instructions and source code have been provided for the following situations:

a) A program that makes files suitable for importing to spreadsheet type programs (e.g., LOTUS-123 and RS/1) and for printing and editing (see Section 7.1);

b) A program that makes a compact (but "formatted") file suitable for transporting data to another computer or between different programming languages (see Section 7.2);

c) FORTRAN Source Code (with all material to read/write data files) that you can adapt to your data processing needs (see Section 7.3);

d) BASIC Source Code (with all material to read/write the formatted files of b above) that you can adapt to your data processing needs (see Section 7.4).

Before you can use these programs, they must be copied from the UnkelScope Offline Utility Disk provided with the software. If the particular program(s) of interest is not on the system, copy it from the Utility disk to your disk.

> If you are using Release 3.00 or later you should read the Release notes on files before reading/using any of the information in this section. These notes are located at the front of the installation section of the Users Guide. The program UTRANS.EXE referred to is located on the Utilites program disk.

7.1 "Ascii" Table Files - Suitable for Printing, Editing and Importing to Spreadsheet and Data-Base Programs

The program BINLOT (Releases Before 3.00) or the program UTRANS (Releases 3.00 and Later) take an UnkelScope Data file in its usual format and can convert it to an ascii file suitable for printing or importing to a spreadsheet or data-base program. This file will be referred to as an ascii table file. Specific instructions for common spreadsheet programs are given later in this section. BINLOT has two inputs: the input and output file names; the program will not allow the input and output file names to be the same. UTRANS is a menu-driven program described in the Release notes in the Installation Guide section of this manual.

The format of the file written is shown in Figure 7.1. The first line is the file label saved when the file was first stored. All alphanumeric information is enclosed in double quotes so it will be properly interpreted by the spreadsheet program.

```
"four channels          1-Jan-1980  0: 9:28"
"#""Time(Secon""Voltage    ""Voltage    ""#""Voltage    ""Voltage    ""Voltage    "
"#""ds)        ""          ""          ""#""          ""          ""          "
   1     .01000     .56152     .01000   1     .01000     .57617     .01000
   2     .02000    1.82617     .02000   2     .02000    1.84570     .02000
   3     .03000    3.09570     .03000   3     .03000    3.10547     .03000
   4     .04000    1.85547     .04000   4     .04000    1.84570     .04000
   5     .05000     .59082     .05000   5     .05000     .57617     .05000
   6     .06000    -.68359     .06000   6     .06000    -.69336     .06000
   7     .07000   -1.95313     .07000   7     .07000   -1.96777     .07000
   8     .08000   -2.75879     .08000   8     .08000   -2.74414     .08000
   9     .09000   -1.47949     .09000   9     .09000   -1.46484     .09000
  10     .10000    -.20508     .10000  10     .10000    -.19043     .10000
  11     .11000    1.06934     .11000  11     .11000    1.07910     .11000
  12     .12000    2.33398     .12000  12     .12000    2.34863     .12000
  13     .13000    2.61719     .13000  13     .13000    2.60254     .13000
  14     .14000    1.35254     .14000  14     .14000    1.33789     .14000
  15     .15000     .08301     .15000  15     .15000     .06836     .15000
  16     .16000   -1.18652     .16000  16     .16000   -1.20117     .16000
"four channels          1-Jan-1980  0: 9:28"
"#""Time(Secon""Voltage    ""Voltage    ""#""Voltage    ""Voltage    ""Voltage    "
"#""ds)        ""          ""          ""#""          ""          ""          "
   1     .01000     .58594     .01000   1     .01000     .60059     .01000
   2     .02000    1.85547     .02000   2     .02000    1.87012     .02000
   3     .03000    3.09082     .03000   3     .03000    3.08105     .03000
   4     .04000    1.83105     .04000   4     .04000    1.82129     .04000
   5     .05000     .56641     .05000   5     .05000     .55176     .05000
   6     .06000    -.70801     .06000   6     .06000    -.71777     .06000
   7     .07000   -1.98242     .07000   7     .07000   -1.99707     .07000
   8     .08000   -2.72949     .08000   8     .08000   -2.71484     .08000
   9     .09000   -1.45020     .09000   9     .09000   -1.43555     .09000
  10     .10000    -.17578     .10000  10     .10000    -.16602     .10000
  11     .11000    1.09375     .11000  11     .11000    1.10352     .11000
  12     .12000    2.36328     .12000  12     .12000    2.37305     .12000
  13     .13000    2.58789     .13000  13     .13000    2.57813     .13000
  14     .14000    1.32813     .14000  14     .14000    1.31348     .14000
  15     .15000     .05859     .15000  15     .15000     .04395     .15000
  16     .16000   -1.21582     .16000  16     .16000   -1.23047     .16000
```

Figure 7.1 "Ascii Table" file format written by program BINLOT.

The second and third lines contain the labels for the data traces. The first 10 characters of each label are on the second line; the second 10 characters are on the third line. The data from each Vertical Trace has 4 columns. Data from Vertical Trace 1 and Vertical Trace 2 are output first. The first column is the sample number. The second column contains the time values and the third column contains the vertical trace values. The final column contains the values stored as the horizontal trace if the data was taken as a cross-plot; if the data was not taken as a cross-plot this column can be ignored. The values of time, vertical trace and horizontal trace are given in decimal format with 5 digits after the decimal point (e.g. , - 1000.17000). The next four columns contain the data for Vertical Trace 2. Each of the four columns is given since a data file may contain data taken from more than one experiment and thus the time values may not be the same as those for Trace 1. The program will display each set of results up to the last trace containing data. Thus if only the first Trace has data, the second set of data is not printed. However, if Trace 3 contained d ata, but Trace 2 did not, results for Trace 2 would still be printed.

The data from Vertical Traces 3 and 4, if any, follows a blank line after the data from the first two Vertical Traces. The format is the same: the first two lines contain the labels and the data follows those lines.

The file size may be quite large. For a full 4 Traces with 1024 points, the data file will be about 160 K bytes or almost half a double density, double sided floppy disk.

7.1.1 Displaying or Printing the Ascii Table File. This file can be displayed to the screen using the DOS TYPE command or printed with the DOS PRINT command. The lines are less than 80 characters long, so they will be printed properly on most screens and printers. Printing the entire file may take a considerable time - there are at least as many lines as data samples.

7.1.2 Editing the Ascii Table Format File. A full file may be very large and cumbersome to work with in an editor but it can be done with most editors. The file length will be 40 x number of samples x highest Trace stored; as mentioned 4 traces with 1024 points gives a file of about 160 K bytes.

7.1.3 Using the Ascii Table File in LOTUS-123.* The file can be transported to LOTUS-123 using the "File" activity and selecting "Import" and then "Numbers" (and quoted strings). This will place data from the latter two traces below the first two traces. You can then move those rows/columns as necessary. The cell sizes may also need to be expanded to see all the digits stored.

*LOTUS-123 is a Trademark of Lotus Development Corporation.

7.2 Ascii Transfer Files - Suitable for Transferring Data to User Written Programs and to other Computers.

UnkelScope files stored on floppy disks can be directly transferred to another computer of the same type and operating system with a compatible floppy disk drive, e.g., between two IBM PC's running DOS. There are two difficulties with transferring the UnkelScope standard 'binary' files between two computers under other circumstances:

a) File transfer protocols for 'serial' or 'parallel' communication lines between computers may not handle the "binary" information. (In Ascii files each character is represented by 7 bits of an 8 bit byte; the eighth bit is processed in a special way. In 'binary' all 8 bits are needed and information is lost when the 8th bit is treated as a special bit); and,

b) Even if a file transfer protocol can handle the 'binary' data, a second problem may arise because the construction of the "file records" are different or because the computers represent numbers in a different fashion. This means that the binary files are not usually transportable from language to language on the same machine.

Thus it is not always possible to directly transfer the binary files between different computers. Rather than attempting to make the binary file transfer between different machines, it is probably better to accept the larger sized ascii files, with their slower transfer rates. The ascii table files described in Section 7.1 can be transported with no problem but the ascii format used by the program discussed here has the advantage of retaining all significant digits of the data with only a modestly increased length. This file can also be printed or examined, but the data is stored differently than the earlier described file. This file is referred to as an ascii transfer file.

The program BINASC (UnkelScope Releases prior to 3.00) or the program UTRANS (UnkelScope Releases 3.00 and Later) will allow you to convert UnkelScope files into an ascii file suitable for transferring from computer to computer. BINASC has two inputs: the input and output file names; the program will not allow the input and output file names to be the same. UTRANS is a menu-driven program described in the Release notes in the Installation Guide section of this manual.

The program ASCBIN (UnkelScope Releases prior to 3.00) or the program UTRANS (UnkelScope Releases 3.00 and Later) will allow you to convert an UnkelScope transfer file into an normal UnkelScope file. ASCBIN has two inputs: the input and output file names; the program will not allow the input and output file names to be the same. UTRANS is a menu-driven program described in the Release notes in the Installation Guide section of this manual.

The Programs described here apply directly only to Releases prior to Release 3.00. If you wish to manipulate UnkelScope data files directly (i.e. without first translating them to Ascii files) for Releases after 3.00, you should contact Unkel Software for assistance.

7.3 Manipulating UnkelScope Data Files Within The FORTRAN Environment

To allow the most flexibility for the FORTRAN programmer, the source code listings are provided for a two "template" program that allows offline manipulation of data from files. The first template program "PROCES" takes an UnkelScope binary data file, performs some simple processing and writes the results to a second UnkelScope binary file. The second template program "INTEGR" shows how analytical results can be stored in a data file that can be plotted using the UnkelScope. These programs are fully functional, but are expected to serve as templates or examples. In cases where statements are compiler dependent (e.g., file opens etc.) the templates use the MICROSOFT V3.2 as a standard. Thus, some modifications may be required if you have compilers made by Digital Research for example.

Modifying and using code written by someone else can be a frustrating experience, because programming styles are so varied. To help make these programs as useful as possible they are described in prose, in flow chart and in comments in the code itself. The code is divided generally into two sections, one of which is common to both template programs; this common section deals mostly with the reading and writing of files and is discussed last.

No discussion of editors or of the detailed procedure for making FORTRAN source code into a program is given. For these, the user is referred to the appropriate manuals. The general process will be to compile each of the source codes and link them all together. Specifically you will need to compile each of the fortran programs FUTILS.FOR, FILIO.FOR, FILIOA.FOR, FILIOT.FOR. Then compile the main program (PROCES for example). Finally, link the object files of all these together to make a complete program.

7.3.1 Template Offline Program "PROCES"

This program takes a file with sampled data of the instantaneous voltage and current to determine the instantaneous power dissipated in a device and the total energy used from the start of the scan. The voltage is stored in Vertical Trace 1 and the current is stored in Vertical Trace 2. Nothing is stored in Vertical Traces 3 or 4. Figures 7.2a and 7.2b show the setup configuration and representative data for this example. The instantaneous power is stored as Horizontal Trace 1 and the total energy is stored as Horizontal Trace 2. This approach (as opposed to using Vertical Trace 3 and Vertical Trace 4) saves space.

```
---------- Vertical Trace 1 ----------      ------ Additional Vertical Traces ------
Source [Analog  0]  A/D Range [+/- 10]      Tr Input                            A/D
Label  :Voltage (volts)                     #  Chan  Label:                   Range
Span   [ 10    v full scale]                3 [none ]
Range  [-5.00E+00 to  5.00E+00]             4 [none ]

---------- Vertical Trace 2 ----------
Source [Analog  1]  A/D Range [+/- 10]
Label  :Current (amps)
Span   [ 10    v full scale]
Range  [  .00E+00 to  1.00E+01]             ------------- Triggering -------------
                                            Mode  [Singl Sweep] Source [Keyboard ]
---------- Horizontal Trace ----------
Source [Time      ]
Label  :Time(Seconds)                       ------------- Processing -------------
Span   [ 1    s full scale]                 Type [none                     ]
Range  [  .00E+00 to  1.00E+00]

-------------- Sampling --------------
Sample Rate [  1 ms        1  khz]
[1024] Samples  (Scan Time 1.02E+00 s)
(Delayed Plot  ) (Processing Inactive)
```

Figure 7.2a Setup screen to take data for example program PROCES.

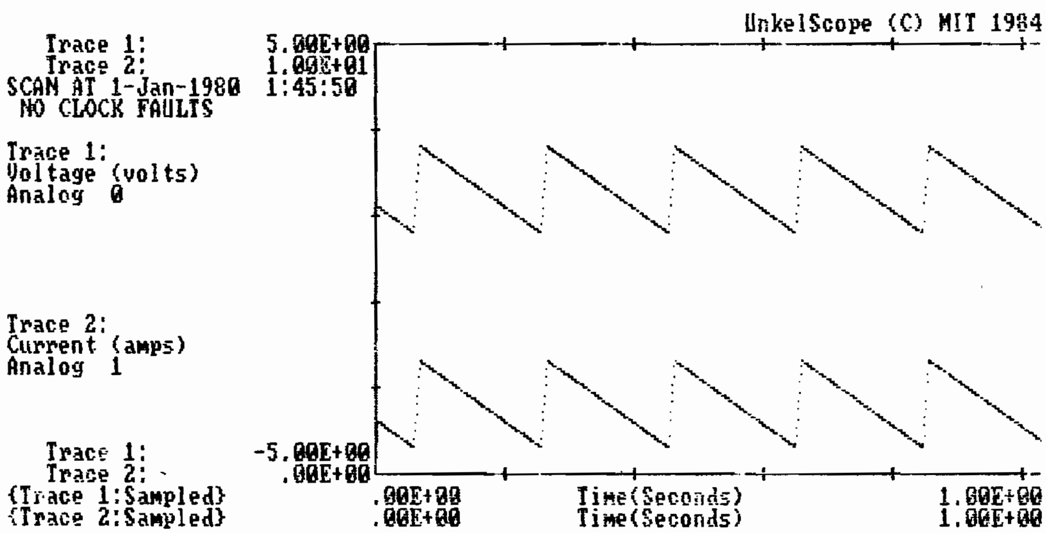


Figure 7.2b Display screen showing data set to be processed by the offline program PROCES.

Figure 7.3 shows a flowchart of the 'PROCES' template program; the code with comments is shown as Figure 7.4. A variety of declarations are made, including the declaring of the usual FORTRAN I-N as INTEGER*2 (not default INTEGER*4). The remaining are individual declarations of character type.

The program starts by initializing a variety of things, including the data arrays and the numbers (LSI03, LSI04) that will be used as logical unit numbers for the file reads and writes. The program then calls a subroutine INNOUT which requests the input and output file names subject to the values stored in integers NIN and NOUT. In this case, NIN and NOUT are each set to 1, indicating to INNOUT that both an input file will be read and an output file will be written. The subroutine INNOUT performs the checks to insure that the input and output file names are not the same. The file names are stored in the labeled common /mnames/ and should not be modified.

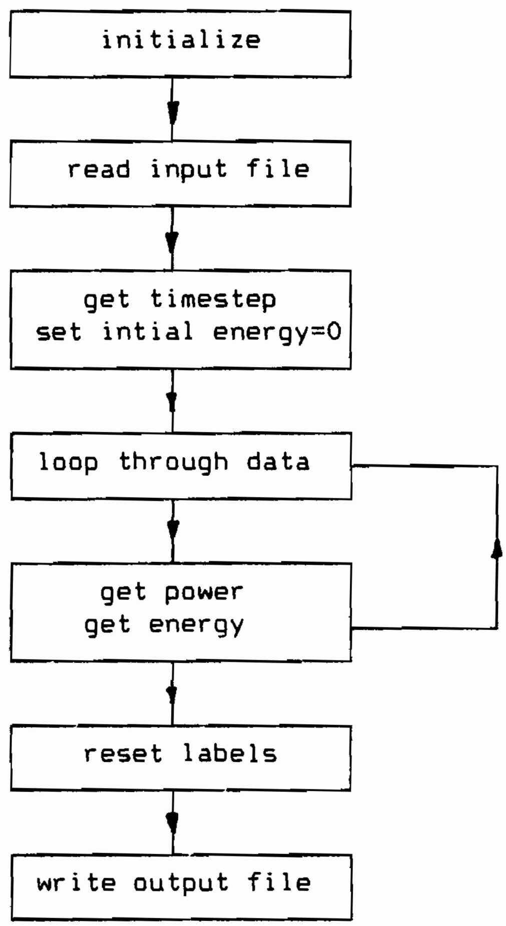


Figure 7.3 Flowchart of program PROCES.

```
c this is the main program for "proces" as described in
c   Section 7.3 of the UnkelScope Guide
c
c this declares the usual integers as *2 instead of *4
           implicit integer*2 (i-n)
c this declares the arrays storing the labels as character*1
        character*1 bpv1la,bph1la,bpt1la,
     $              bpv2la,bph2la,bpt2la,
     $              bpv3la,bph3la,bpt3la,
     $              bpv4la,bph4la,bpt4la,
     $              bhead1,bhead2
c this common holds the actual data for each of the traces.
           COMMON/MDATAF/
     $ VERT1(1024),HORZ1(1024),TIME1(1024),
     $ VERT2(1024),HORZ2(1024),TIME2(1024),
     $ VERT3(1024),HORZ3(1024),TIME3(1024),
     $ VERT4(1024),HORZ4(1024),TIME4(1024)
c this common holds the number of channels and the number of
c       data points in each trace
           common/mdatan/nchd,npt1,npt2,npt3,npt4
c this common holds the labels for the traces
c       for example bpv1la is the label for vertical trace 1
           common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),
     $ bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),
     $ bpv4la(20),bph4la(20),bpt4la(20)
           common/mdatah/
     $ bhead1(40),bhead2(40)
c this common holds numbers used for input and output. DONOT change them
        common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c this common holds the names used to access the data files. DONOT change them
        common/mnames/bfinp(10),bfout(10)
c
c now starts the program
c
c setup initial values
         call init(1024)
c get input and output file names as needed
c        codes for nin and nout:
c          0 - dont want any
c          1 - binary file
c          2 - ascii file for transfer
c          3 - ascii file suitable for printing or porting to lotus etc.
c for this case set nin=1 to read binary file as input
         nin = 1
c for this case set nout=1 to write binary file
         nout = 1
         call innout(nin,nout)
```

Figure 7.4 Source listing for FORTRAN program PROCES.
(Part 1 of 2)

```
c if there is an input file open it and read the file
c       routine binput prints out certain things about the file
c
        if(nin .gt. 0)call binput(nin,maxch,maxlen,ner)
c
c
c this is the part of the file that actually processes the data
c   determine the time step assuming that points are evenly spaced
       dtime=time1(2)-time1(1)
c   set the initial energy to zero
       energy=0.0
c  do loop to do processing for all points
       do 50 i=1,npt1
c  compute the power for point i
c   power is stored in kilowatts
       power=vert1(i)*vert2(i)/1000.
c  add the energy in the past time step
       energy=energy+power*dtime
c  now store values into the arrays horz1 and horz2
       horz1(i)=power
       horz2(i)=energy
c  the end of the loop
50     continue
c now to get the labels set properly
c            length   array    quoted string
c
       call bstuf(20,bph1la,'   Power (kwatts)    ')
       call bstuf(20,bph2la,'  Energy (kJoules)   ')
c
c if there is an output file, open it and write the file
c    nout=1 means that a binary file will be written
        if(nout .gt. 0)call boutpu(nout,maxch,maxlen,ner)
c
c program stops
        stop
        end
```

Figure 7.4 Source listing for FORTRAN program PROCES.
(Part 2 of 2)

The next statement calls subroutine BINPUT to open, read and close an UnkelScope file. The variable NIN is used to designate one of the valid input file types; in this case with NIN=1 a binary file is read. The subroutine writes the file label to the screen; if the file cannot be found a message is printed and the program stops. The variable MAXCH returns the number of vertical traces that may have data; for this example the expectation is that MAXCH=2. MAXLEN is the length of the longest set of Vertical Trace data; in this example both should be the same size and MAXLEN=1024. The main result of reading the file is to fill information into the data commons: mdataf, mdatan, mdatal and mdatah. A description of each common as it relates to this example is given in the section below.

mdatah. This holds two 40 character labels. The first label identifies the file as a valid UnkelScope file; you should not alter this label. The second label (bhead2) holds the label stored with the file.

mdatal. This holds the 20 character labels identifying traces. For example, bpV1la will hold the string "Voltage(volts)" and bpV2la the string "Current(amps)." The two time labels bpT1la and bpT2la will both hold the string "Time(seconds)." The labels bpH1la and bpH2la will be altered later to reflect the information put there.

mdatan. This holds the number of valid samples in each of the Traces stored; NPT1 and NPT2 store the number of points in Traces 1 and 2 respectively and should both be 1024 in this example.

mdataf. The arrays in this common hold the data itself. VERT#, HORZ# and TIME# store respectively, the vertical, horizontal and time data values for Trace #. VERT1(i) should store the voltage and VERT2(i) store the current at time. TIME1(i)=TIME2(i). The power and energy will be put into HORZ1 and HORZ2 respectively.

The next step is the actual processing. The power is computed (p=v i) and stored in HORZ1. The energy is found by adding to the previous energy the product of the power and the time increment; the initial energy is assumed to be zero, and all time increments the same.

The horizontal trace labels are then modified to be consistent with the information now stored there. The program BSTUF is used to fill the arrays with desired string. Subroutine BSTUF takes three arguments; the first is the number of characters (set to 20), the second is the name of the array to be filled, and the third argument is the quoted string with the label (exactly 20 characters or blanks must be contained between the double quotes). If the number of Traces was altered, MAXCH would need to be altered.

The final step is to open the output file, write the data and close the file. This is done with the call to BOUTPU. The value NOUT=1 indicates that the file is to be an UnkelScope binary file.

The program then quits. The screen output from running the program is shown in Figure 7.5a. The setup configuration and the plotted power and energy are shown in Figures 7.5b and c.

```
C>proces
 Input file name(10 characters;no blanks)?ramps.dat
Output file name(10 characters;no blanks)?ramps.pow
Label from Input File:device performance   1-Jan-1980  1:54:44
there are 2 channels in this file
Stop - Program terminated.
```

Figure 7.5a Screen dialog for program PROCES.

```
---------- Vertical Trace 1 ----------        ------ Additional Vertical Traces ------
Source [File Hor1]File Name:ramps.pow          Tr Input                          A/D
Label  :   Power (kwatts)                      #  Chan  Label:                 Range
Span   [50   mv full scale]                    3 [none ]
Range  [-2.50E-02 to  2.50E-02]                4 [none ]

---------- Vertical Trace 2 ----------
Source [File Hor2]File Name:ramps.pow
Label  :  Energy(kJoules)
Span   [10   mv full scale]
Range  [  .00E+00 to  1.00E-02]               ------------ Triggering ------------
                                               Mode  [Singl Sweep] Source [Keyboard ]
---------- Horizontal Trace ----------
Source [Time      ]                            ------------ Processing ------------
Label  :Time(Seconds)                          Type [none                      ]
Span   [ 1   s full scale]
Range  [  .00E+00 to  1.00E+00]

-------------- Sampling --------------
Sample Rate [  1 ms       1  khz]
[1024] Samples  {Scan Time 1.02E+00 s}
{Delayed Plot  } {Processing Inactive}
```

Figure 7.5b Setup configuration to retrieve data processed by program PROCES.

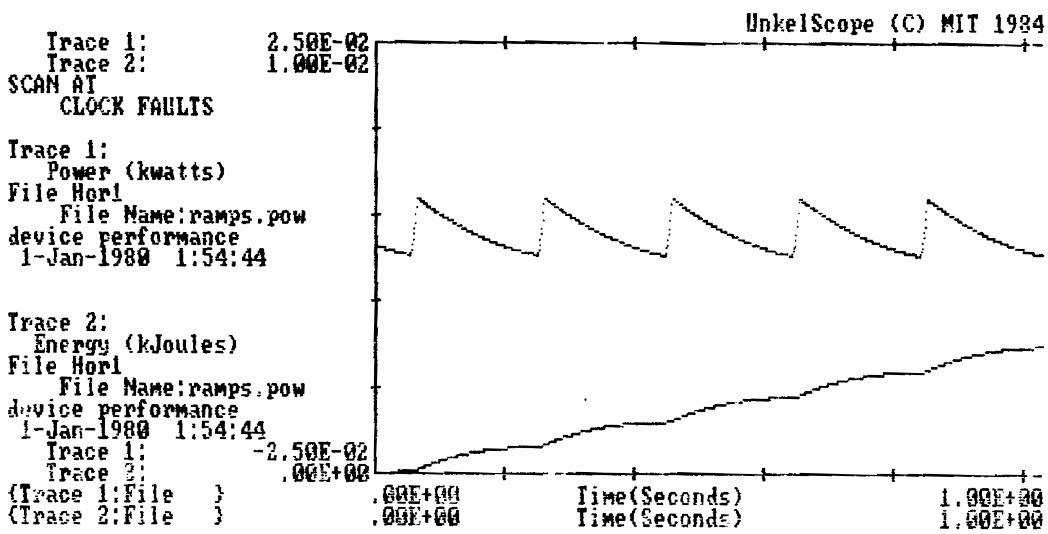


Figure 7.5c Display screen of processed data by program PROCES.

7.3.2 Template Offline Program INTEGR

This program solves the equations of motion for a ball dropped from a building. The ball is accelerated by gravity and slowed by the drag force of the air on it. The situation is shown schematically in Figure 7.6 with the force balance and resulting equation of motion derived. The second- order differential equation is solved by first breaking it up into two coupled first-order equations in the unknowns position Z and velocity V. These equations are written in the form:

$$\frac{dV}{dt} = -g + C_D 1/2\ r\ V^2\ A/m$$

$$\frac{dZ}{dt} = V$$

where g is the acceleration of gravity , C_D is the drag coefficient, r the density of air, m the mass of the ball and A the projected area of the ball ($=\pi R^2$ where R is the radius of the ball.) The solution is obtained by numerically integrating the equations from the initial conditions

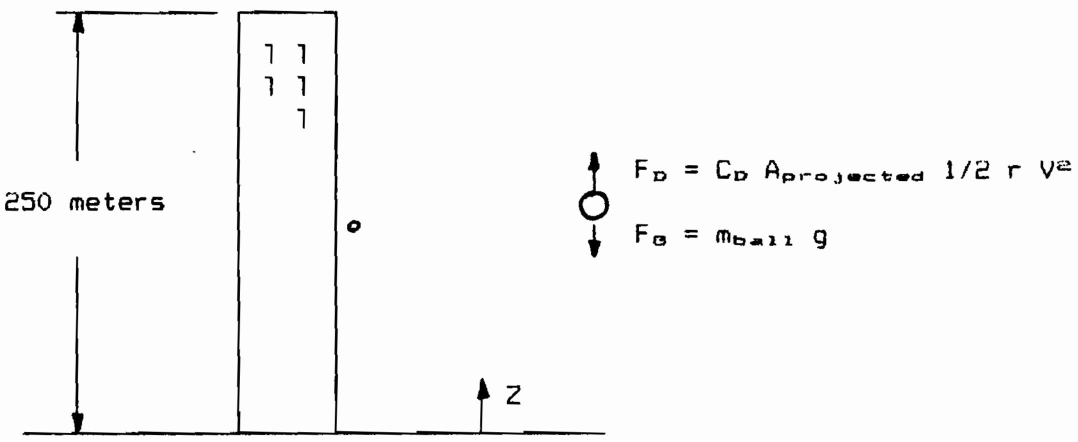
Z = 250 meters at t = 0 seconds and
V = 0 meters/sec at t = 0 seconds

The time step of integration is set at approximately 1/1000th of the total time for which the equations are integrated.

Integration of the equations should show that the ball reaches a 'terminal velocity' with value

$$V_{terminal} = 2mg/C_D\ r\ A$$

and this value can be compared with the value predicted by numerical integration.



The drag force F_D on the ball can be calculated as

$$F_D = C_D A_{projected} 1/2 r V^2$$

where C_D is the drag coefficient (formally a function of the Reynolds Number (rVD/m), but assumed to be a constant =1.2 for this example), $A_{projected}$ is the projected area = πr_{ball}^2, r is the density of air (=1.1 kg/m3) and V is the velocity of the ball measured with positive velocity upward. The force of gravity F_G on the ball can be determined as

$$F_G = m_{ball} g$$

where m_{ball} is the mass of the ball, and g is the acceleration of gravity.

The force balance on the ball yields,

$$m_{ball} a = m_{ball} dV/dt = m_{ball} d^2Z/dt^2 = F_D - F_G$$
$$= C_D A_{projected} 1/2 r V^2 - m_{ball} g$$

where a is the acceleration of the ball, Z is the position of the ball and t is time.

The equivalent representation of the equation of motion as a pair of coupled first order equations is

$$dV/dt = C_D A_{projected} 1/2 r V^2 - m_{ball} g$$

$$dZ/dt = V$$

Figure 7.6 Sketch of the physical problem and derivation of the equations solved by the program INTEGR.

The program INTEGR is set-up to read values for the ball mass, the ball radius and the total integration time. The drag coefficient, density of air and gravitational constants ar set in the program. Th computed values of position an velocity are stored in the arrays VERT1 and VERT2 respectively.

Figure 7.7 shows a flow chart of the template program INTEGR the code with comments is shown as Figure 7.8. A variety of declarations are made, including the declaring of th usual FORTRAN I-N as INTEGR*2 (not the default INTEGR*4). The remaining are individual declaration of character type.

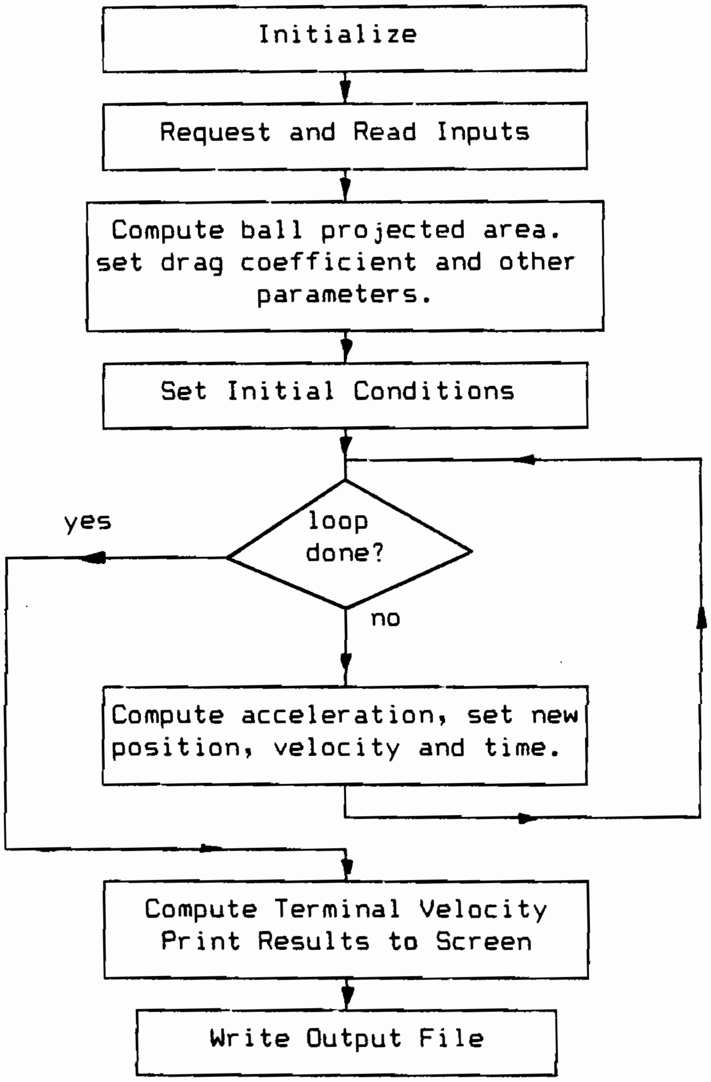


Figure 7.7 Flow chart of program INTEGR.

```
c this is the main program for "integr" as described in
c   Section 7.3 of the UnkelScope Guide
c
c this declares the usual integers as *2 instead of *4
          implicit integer*2 (i-n)
c this declares the arrays storing the labels as character*1
       character*1 bpv1la,bph1la,bpt1la,
     $             bpv2la,bph2la,bpt2la,
     $             bpv3la,bph3la,bpt3la,
     $             bpv4la,bph4la,bpt4la,
     $             bhead1,bhead2
c this common holds the actual data for each of the traces.
          COMMON/MDATAF/
     $ VERT1(1024),HORZ1(1024),TIME1(1024),
     $ VERT2(1024),HORZ2(1024),TIME2(1024),
     $ VERT3(1024),HORZ3(1024),TIME3(1024),
     $ VERT4(1024),HORZ4(1024),TIME4(1024)
c this common holds the number of channels and the number of
c      data points in each trace
          common/mdatan/nchd,npt1,npt2,npt3,npt4
c this common holds the labels for the traces
c      for example bpv1la is the label for vertical trace 1
          common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),
     $ bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),
     $ bpv4la(20),bph4la(20),bpt4la(20)
          common/mdatah/
     $ bhead1(40),bhead2(40)
c this common holds numbers used for input and output. DONOT change them
      common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c this common holds the names of the data files. DONOT change them
       common/mnames/bfinp(10),bfout(10)
c
c now starts the program
c
c setup initial values
        call init(1024)
c
7000    continue
c request and read the values of ball mass and radius
c
        write(*,6000)

c the $ at end of the format leaves cursor at the same line
6000    format(' Program to Solve for Motion of Dropped Ball'/)
701     continue
        write(*,600)
600     format(' Input Ball Mass (kg):',$)
        read(*,*)bmass
```

Figure 7.8 Program listing for INTEGR. (Part 1 of 3.)

```
c check for reasonable mass
        if(bmass .gt. 0.0)go to 700
        write(*,601)
601     format(' Improper ball mass. Try again.')

        go to 701
700     continue
        write(*,602)
602     format(' Input Ball Radius (meters):',$)
        read(*,*)bradi
        if(bradi .gt. 0.0)go to 702
        write(*,6002)
6002    format(' Improper ball radius. Try again.')
        go to 700
702     continue
        write(*,6001)
6001    format(' Input Total Time (seconds):'$)
        read(*,*)tfinal
        if(tfinal .ge. 0.0)go to 7002
        write(*,6003)
6003    format(' Improper total time. Try again.')
        go to 702
7002    continue
c now compute the projected area of the ball
        area=3.14159*bradi*bradi
c now set the other parameters
        cdrag=1.2
        grav=9.81
        rhoair=1.17
c now compute the time step to be used
        dtime=tfinal/1024
c get input and output file names as needed
c       codes for nin and nout:
c         0 - dont want any
c         1 - binary file
c         2 - ascii file for transfer
c         3 - ascii file suitable for printing or porting to lotus etc.
c for this case set nin=0 because there is on input file

        nin = 0
c for this case set nout=1 to write binary file
        nout = 1
        call innout(nin,nout)
c if there is an input file open it and read the file
c       but with nin=0 nothing will be read
        if(nin .gt. 0)call binput(nin,maxch,maxlen,ner)
c now set the initial conditions
        posit=250.
        veloc=0.0
        time=0.0
```

Figure 7.8 Program listing for INTEGR. (Part 2 of 3.)

```
c now the integration loop
        do 50 i=1,1024
c time
        time=time+dtime
c position
        posit=posit+veloc*dtime
c velocity - compute the acceleration
        accel=-grav+0.5*cdrag*rhoair*veloc*veloc*area/bmass
        veloc=veloc+accel*dtime
c now store the values into the data arrays
        vert1(i)=posit
        vert2(i)=veloc
        time1(i)=time
        time2(i)=time
c end of the loop
50      continue
c compute terminal velocity
        term=-sqrt(2.*bmass*grav/(cdrag*rhoair*area))
c now print final velocity and terminal velocity
        write(*,6006)posit,veloc,time,term
6006    format(' final position = ',1pe10.3,';final velocity = ',1pe10.3,
     $  ' at time = ',1pe10.3/
     $  ' computed terminal velocity = ',1pe10.3)
c now to get the labels set properly
c file
       call bstuf(40,bhead2,
     $   '   Ball Dropped From Tall Building      ')
c traces      length   array    quoted string
c
       call bstuf(20,bpv1la,'    Position(m)     ')
       call bstuf(20,bpv2la,'   Velocity(m/s)    ')
       call bstuf(20,bpt1la,'    Time(seconds)   ')
       call bstuf(20,bpt2la,'    Time(seconds)   ')
c set horizontal labels to garbage
       call bstuf(20,bph1la,'     GARBAGE        ')
       call bstuf(20,bph2la,'     GARBAGE        ')
c now set maximum traces to 2
       maxch=2
c set maximum number of points of trace 1 and 2
       npt1=1024
       npt2=1024
c set largest length of the traces
       maxlen=1024
c
c if there is an output file, open it and write the file
c    nout=1 means that a binary file will be written
        if(nout .gt. 0)call boutpu(nout,maxch,maxlen,ner)
c
c program stops
        stop
        end
```

Figure 7.8 Program listing for INTEGR. (Part 3 of 3.)

The program starts by initializing a variety of things, including the data arrays and numbers (LSI03, LSI04) that are used as logical unit numbers for file reads and writes.

The next step is to write a prompt to the screen to request the values for ball mass, radius and total time. The ",$" at the end of the formats printing the prompts suppresses the carriage return and line feed so that the inputs are typed on the same line as the prompt. This feature may not work in all implementations of Fortran. The values are read from the keyboard using "free format" input. In MicroSoft Fortran the screen/keyboard is accessed as logical unit *.

The program then initializes the values of velocity and position and determines the timestep. The program then calls a subroutine INNOUT (described in Section 7.3.3) which requests the input and output names subject to the values stored in integers NIN and NOUT. In this case NIN is set to 0 to indicate that no input file is to be read. NOUT is set to 1 to indicate that output file will be written. Subroutine INNOUT takes care of getting the file names stored in the labeled common /mnames/; variables in this common should not be modified.

The program then integrates the equations, storing the results in arrays VERT1 and VERT2 and the time in TIME1 and TIME2. When the final time is reached the program computes the terminal velocity. The final velocity of the integration and the terminal velocity computed by the analytical equation are printed to the screen. The variable MAXCH is set equal to 2 (VERT1 and VERT2 contain results) and MAXLEN is set equal to 1024.

The program must then set the values of variables stored in the Labeled commons. A description of each common and the action taken to fill it properly is given below.

mdatah. This holds two 40 character labels. The first label (bhead1) will be constructed by the program that writes the output file; you should not alter this label. The second label (bhead2) is modified to read "Ball Dropped From Tall Building". The program BSTUF is used to fill this array with the desired string. Usage of BSTUF is described in Section 7.3.3.

mdatal. This holds the 20 character labels identifying traces. The Vertical Trace labels bpV11a is altered to read "Position(m)" and the label bpV21a is altered to read "Velocity(m)". The time trace labels bpT11a and bpT21a are both altered to read "Time(seconds)". The labels for the Horizontal Traces bpH11a and bpH21a are set to "garbage" since nothing has been stored there.

mdatan. This holds the number of valid samples in each of the traces stored; NPT1 and NPT2 are both set to 1024.

mdataf. These arrays hold the actual values and have already been set as appropriate.

The final step is to open the output file, write the data and close the file. This done with the call to Subroutine BOUTPU. The value NOUT=1 indicates that the file is to be an UnkelScope binary file.

The program then quits. The screen dialog while running the program is shown in Figure 7.9a. The setup configuration and plotted results are shown in Figures 7.9b and 7.9c respectively.

```
C>integr
Program to Solve for Motion of Dropped Ball

Input Ball Mass (kg):0.09
Input Ball Radius (meters):0.0254
Input Total Time (seconds):10.
Output file name(10 characters;no blanks)?ball.dat
final position =  4.477E+01;final velocity = -2.489E+01 at time =  1.000E+01
computed terminal velocity = -2.491E+01
```

Figure 7.9a Program INTEGR screen input and output.

```
------------ Vertical Trace 1 ------------   -------- Additional Vertical Traces --------
Source [File Tr 1]File Name:ball.dat         Tr Input                              A/D
Label  :    Position(m)                      #  Chan  Label:                      Range
Span   [500   v full scale]                  3 [none ]
Range  [-2.50E+02 to  2.50E+02]              4 [none ]

------------ Vertical Trace 2 ------------
Source [File Tr 2]File Name:ball.dat
Label  :   Velocity(m/s)
Span   [ 50   v full scale]
Range  [-2.50E+01 to  2.50E+01]              --------------- Triggering ---------------
                                             Mode  [Auto        ]
------------ Horizontal Trace ------------
Source [Time      ]
Label  :Time(Seconds)                        --------------- Processing ---------------
Span   [10    s full scale]                  Type [none                     ]
Range  [  .00E+00 to  1.00E+01]

---------------- Sampling ----------------
Sample Rate [ 50 ms        20 hz]
[1024] Samples  {Scan Time 5.12E+01 s}
{Real Time Plot} {Processing Active  }
```

Figure 7.9b Setup configuration to display results for program INTEGR.

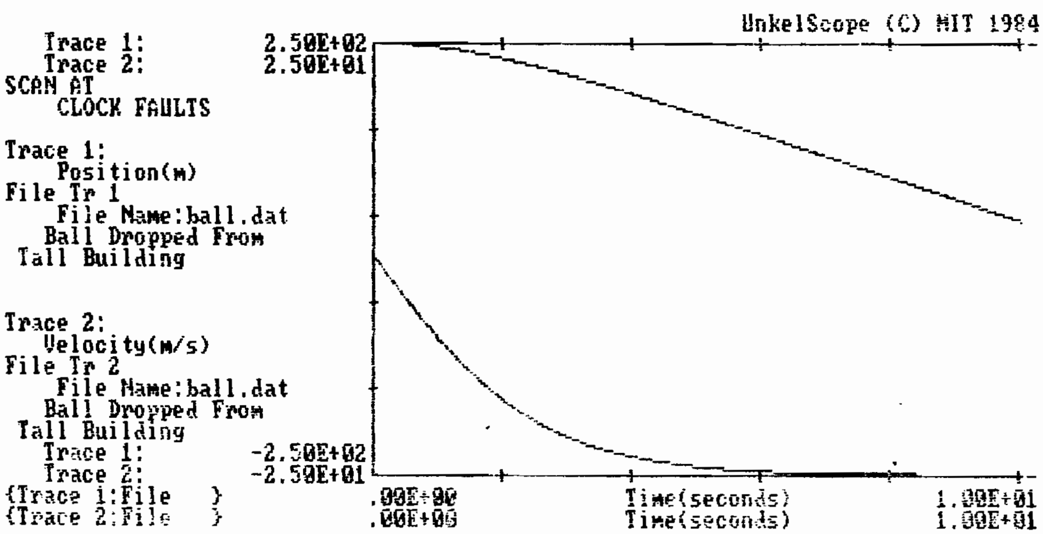


Figure 7.9c Plotted results for program INTEGR.

7.3.3 Utilities for Offline Manipulation Under FORTRAN

The previous programs 'PROCES' and 'INTEGR' make calls to several subroutines that generally deal with reading writing the files, modifying labels and requesting file names. In the section below each of the subroutines is described briefly and the program listings given. Emphasis is on the correct usage of the subroutine rather than describing in detail the program statements, since the sub-programs should not be modified.

The programs are organized into 4 source files:

FUTILS.FOR
FILIO.FOR
FILIOT.FOR
FILIOA.FOR

7.3.3.1 FUTILS.FOR

SUBROUTINE INIT(NPONTS). This subroutine initializes the data arrays (to 0.0) and sets a variety of other values needed. The subroutine takes 1 input NPONTS which should be 1024 for this version. The code listing with comments is shown as Figure 7.10.

SUBROUTINE INNOUT(NIN,NOUT). This routine requests input and output file names as specified by the integers NIN and NOUT. An input name is requested if NIN is greater than 0; an Output file name is requested if NOUT is greater than 0. If both input and output file are requested, the program will not allow these files to have the same name. The file names are stored in the variables bfinp and bfont located in common/mnames/. They should not be altered. The code listing with comments is shown as Figure 7.11.

SUBROUTINE BINPUT(NIN,MAXCH,MAXLEN,NER). This routine coordinates the reading of the input file by opening the file, calling the proper subroutine to read it and handling errors. NIN determines what type of file should be read. If NIN is 1 the file is supposed to be a binary file; the file is opened as binary and the routine RETFIL called to read it. If NIN is 2 the file is supposed to be an ascii transfer file (see Section 7.2) and the subroutine RETASC is called to read it. If NIN is 3 the file is opened, but this option is not supported in that when subroutine RETTAB is call ed the program will stop on an error. The first 40 characters of the file are used to identify the file as a valid file.

The variables MAXCH and MAXLEN are returned by the program reading the file and are respectively the highest Trace number stored in the file and the longest length of a data array. NER is an error code returned from the subroutine reading the file. If NER=0, all is well; otherwise the program will quit.

The code listing with comments is shown in Figure 7.12

```
c this routine initializes various things
      subroutine init(nponts)
c declare usual integers to integer*2
      implicit integer*2 (i-n)
c declare variable character*1 type
       character*1 bfinp,bfout
c this common holds all the data
           COMMON/MDATAF/
     $ VERT1(1024),HORZ1(1024),TIME1(1024),
     $ VERT2(1024),HORZ2(1024),TIME2(1024),
     $ VERT3(1024),HORZ3(1024),TIME3(1024),
     $ VERT4(1024),HORZ4(1024),TIME4(1024)
c this holds the lengths of the data arrays
       common/mdatan/nchd,npt1,npt2,npt3,npt4
c this holds the names of the input and output files
       common/mnames/bfinp(10),bfout(10)
c this holds numbers used in file reads and writes
       common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c
c zero the arrays
       do 509 i=1,nponts
       vert1(i)=0.0
       vert2(i)=0.0
       vert3(i)=0.0
       vert4(i)=0.0
       horz1(i)=0.0
       horz2(i)=0.0
       horz3(i)=0.0
       horz4(i)=0.0
       time1(i)=0.0
       time2(i)=0.0
       time3(i)=0.0
       time4(i)=0.0
509    continue
        do 588 i=1,10
        bfinp(i)=' '
        bfout(i)=' '
588     continue
        npt1=nponts
        npt2=0
        npt3=0
        npt4=0
        nchd =1
        lsio1=1
        lsio2=1
        lsio3=1
        lsio4=1
        return
        end
```

Figure 7.10 Source code listing for subroutine INIT.

```
c this routine requests the input and output data file names
c   according to the values of nin and nout
      subroutine innout(nin,nout)
         implicit integer*2 (i-n)
         character*1 bfinp,bfout
         common/mdatan/nchd,npt1,npt2,npt3,npt4
         COMMON/MDATAF/
     $ VERT1(1024),HORZ1(1024),TIME1(1024),
     $ VERT2(1024),HORZ2(1024),TIME2(1024),
     $ VERT3(1024),HORZ3(1024),TIME3(1024),
     $ VERT4(1024),HORZ4(1024),TIME4(1024)
c this common will be filled with the file names
      common/mnames/bfinp(10),bfout(10)
c this common holds values used for input and output files
      common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c this routine reads the input and output file names as needed
c       nin = 0 ; no input file.        nout = 0 ; no output file.
c       nin = 1 ; binary input file.    nout = 1 ; binary output file.
c       nin = 2 ; ascii transfer file. nout = 2 ; ascii transfer file.
c       nin = 3 ; not valid.            nout = 3 ; ascii table file.
600     format(80a1)
6999    continue
c request input file name
        if(nin .eq. 0) go to 7000
        write(*,60)
60      format(
     $ '  Input file name(10 characters;no blanks)?',$)
        nn=0
        read(*,600)bfinp
7000    continue
c request output file name
        if(nout .eq. 0)go to 7001
        write(*,61)
61      format(
     $ ' Output file name(10 characters;no blanks)?',$)
        nn=0
        read(*,600)bfout
7001    continue
c this is a trap to prevent writing over the input file
c need to do it only if both are read
        if(nin .eq. 0 .or. nout .eq. 0)go to 7002
        do 510 i=1,10
        if(bfout(i) .ne. bfinp(i)) go to 7002
510     continue
c same name request names again
        write(*,62)
62      format(
     $ '***ERROR:input and output file names the same. Try again!')
        go to 6999
7002    continue
        return
        end
```

Figure 7.11 Source code listing for subroutine INNOUT.

```
c this routine coordinates the reading of the input file
c   and calls the appropriate subroutine to read the proper type file
       subroutine binput(nin,maxch,maxlen,ner)
c declare all integers of length *2
          implicit integer*2 (i-n)
c declare of character*1 type
          character*1 bfout,bfinp,bhead1,bhead2
c this common will be filled with the file names
       common/mnames/bfinp(10),bfout(10)
c this common holds values used for input and output files
       common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c this common holds the heading labels for the file
       common/mdatah/bhead1(40),bhead2(40)
c
c if nin is 1 then file is a binary file
c if nin is 2 or more then file is an ascii file
c
        if(nin .eq. 1)call uopen(lsio3,bfinp,4,ner)
        if(nin .gt. 1)call uopen(lsio3,bfinp,1,ner)
c error message
        if(ner .eq. 0)go to 7000
        write(*,63)
63      format(
     $ '***ERROR:error in file open. Program Exits.                 ')
        stop
c
7000    continue
c
c now the actual read of the file
        if(nin .eq. 1)call retfil(maxch,maxlen,ner)
        if(nin .eq. 2)call retasc(maxch,maxlen,ner)
c note that rettab will simply return an error because
c      program will not accept an ascii "table" (lotus...) file as input.
        if(nin .eq. 3)call rettab(maxch,maxlen,ner)
c now close the file
        call uclose(lsio3,ner)
c now deal with errors
        if(ner .eq. 0)go to 7001
        write(*,64)
64      format(
     $ '***ERROR:error in file read. Program Exits.                 ')
        stop
7001    continue
c write label to screen as diagnostic
        write(*,65)bhead2
65      format(  ' Label from Input File:',40a1)
        write(*,66)maxch
66      format(' there are ',i1,' channels in this file')
c done so return
        return
        end
```

Figure 7.12 Source code listing for subroutine BINPUT.

SUBROUTINE BOUTPU(NOUT,MAXCH,MAXLEN,NER). This subroutine coordinates the writing of the output file by opening the file, calling the proper routine to write it and handling errors. NOUT determines what type of file will be written. If NOUT is 1 the file will be a binary file; the file is opened as a binary file and subroutine STOFIL used to write it. If NOUT is 2 the file will be an ascii transfer file (see Section 7.2) and subroutine in STOASC is used to write it. If NOUT is 3 the file will be an ascii table file (see Section 7.1) and subroutine STOTAB will be used to write it.

The variables MAXCH and MAXLEN must be set properly before the call to the subroutine MAXCH is the largest trace number to be written. It must be less than or equal to 4. MAXLEN is the longest length of the trace data to be written and for the version must be less than or equal to 1024.

The code listing with comments is shown as Figure 7.13.

```
c this routine takes care of writing the files
c   it choses the proper subroutine for the desired type
       subroutine boutpu(nout,maxch,maxlen,ner)
          implicit integer*2 (i-n)
          character*1 bfinp,bfout,bhead1,bhead2
          common/mdatan/nchd,npt1,npt2,npt3,npt4
c this common will be filled with the file names
       common/mnames/bfinp(10),bfout(10)
c this common holds values used for input and output files
       common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c this common holds the file labels
       common/mdatah/bhead1(40),bhead2(40)
c if nin is 1 then file is a binary file
c    otherwise the file is an ascii file
        if(nout .eq. 1)call uopen(lsio4,bfout,5,ner)
        if(nout .gt. 1)call uopen(lsio4,bfout,2,ner)
c error message
        if(ner .eq. 0)go to 7000
        write(*,63)
63      format('***ERROR:error in file open. Program Exits')
        stop
7000    continue
c now the actual read of the file
        if(nout .eq. 1)call stofil(maxch,maxlen,ner)
        if(nout .eq. 2)call stoasc(maxch,maxlen,ner)
        if(nout .eq. 3)call stotab(maxch,maxlen,ner)
c now close the file
        call uclose(lsio4,ner)
c now deal with errors
        if(ner .eq. 0)go to 7001
        write(*,64)
64      format('***ERROR:error in file read. Program Exits')
        stop
7001    continue
        return
        end
```

Figure 7.13 Source code listing for subroutine BOUTPU.

SUBROUTINE BSTUF (N,BARRAY,B). This subroutine will take the N characters contained between single quotes and stuff them into the character array BARRAY. There must be exactly N characters between the single quotes that delimit the input variable B in the call statement.

For example:

```
          CALL BSTUF (10,BJUNK,'1234567890')
```

The code listing with comments for BSTUF is shown as Figure 7.14.

```
c this routine allows quoted strings to be put into arrays
c   used for filling the label arrays
c
          SUBROUTINE BSTUF(N,BARRAY,B)
c normal integers forced to be integer*2
          implicit integer*2 (i-n)
c character declarations
          character*1 barray,b
C
          DIMENSION BARRAY(1),B(1)
          DO 50 I=1,N
          BARRAY(I)=B(I)
50        CONTINUE
          RETURN
          END
```

Figure 7.14 Source code listing for subroutine BSTUF

SUBROUTINE UOPEN(LU,BD,IAGE,NER). This routine handles the opening of files with the proper attributes and handles error checking. The variable LU is the logical unit number for which will be used for the file. The variable BD is a character array holding the name of the file to be opened; in these situations either bfinp or bfout (from common/mnames/) will be passed to BD. The variable IAGE determines whether the file is a new or old file and whether the file is ascii or binary.

The program first does some checking for valid name then uses the FORTRAN INQUIRE statement to find out if the file exists. If the file is supposed to exist (i.e., it is old) and the inquire statement indicates it doesn't exist the program returns NER=1 and does not attempt to open the file. Otherwise the file is opened and if the open is successful the variable NER is returned as 0.

The code listing for UOPEN is shown as Figure 7.15.

```
          SUBROUTINE UOPEN(LU,BD,IAGE,NER)
c declare variables starting with b as character type
          implicit character*1 (b)
          implicit integer*2 (i-n)
          logical lfile
          DIMENSION BD(1),b10(10)
          CHARACTER*10 C10
          equivalence (c10,b10(1))
c this section performs some checks to see if file name is valid.
C pc  checks if file name is 8 characters or less before the .
          NER=1
c check for a dot
          idot=0
          call bstuf(1,bdot,'.')
          do 550 i=1,10
          if(bd(i) .eq. bdot)idot=i
550       continue
c find last non-blank character
          call bstuf(1,bblnk,' ')
          do 551 i=1,10
          ilast=11-i
          if(bd(ilast) .ne. bblnk)go to 755
551       continue
755       continue
c section to check cases with a .
          if(idot .eq.0)go to 756
c cant be more than 8 characters before the .
          if(idot .gt. 9)go to 7000
c cant be more than 3 characters after the .
          if((ilast-idot) .gt. 3)go to 7000
          go to 757
756       continue
c section to check cases with no .
c  cant be more than 8 characters
          if(ilast .gt. 8)go to 7000
757       continue
c microsoft open wants character string not array of character*1
c other checks and stuff into long character array
          ICNT=0
          DO 50 I=1,10
          IF(BD(I) .NE. ' ')ICNT=ICNT+1
          b10(i)  =BD(I)
50        CONTINUE
         IF(ICNT .EQ. 0) GO TO 7000
         IF(BD(1) .EQ. ' ')GO TO 7000
c now use inquire to find out if file exists
         inquire(FILE=c10,EXIST=lfile)
c now if file is old (iage=4 or 1) and file doesnt exist trap as error
         if(iage .ne. 4 .and. iage .ne. 1)go to 7800
        if(lfile)go to 7801
         go to 7000
```

Figure 7.15 Source code listing for subroutine UOPEN. Part 1 of 2.

```
7800    continue
c checks in case file is supposed to be new
c none for now - allow it to write over old file
7801    continue
c okay file as best as can tell

c now open it
c first two are for unformatted (binary) files
        IF(IAGE .EQ. 5)OPEN(lu,file=c10,STATUS='new',
     $ FORM='UNFORMATTED')
        IF(IAGE .EQ. 4)OPEN(lu,file=C10,STATUS='OLD',
     $ FORM='UNFORMATTED')
c these are for formatted (ascii) files
        IF(IAGE .EQ. 2)OPEN(lu,FILE=C10,STATUS='NEW')
        IF(IAGE .EQ. 1)OPEN(lu,FILE=c10,status='old')
        IF(IAGE .EQ. 0)OPEN(lu,file=c10,status='new')
        NER=0
7000    CONTINUE
        return
        END
```

Figure 7.15 Source code listing for subroutine UOPEN.
Part 2 of 2.

SUBROUTINE UCLOSE(LU,NER). This routine closes an open file. LU is the logical unit number of the file to be closed. NER is returned as 0 unless an error was detected.

The code listing for UCLOSE is shown as Figure 7.16.

```
c this routine closes files
          SUBROUTINE UCLOSE(LU,NER)
          implicit integer*2 (i-n)
          NER=1
          CLOSE(lu,status='keep')
          NER=0
7000      CONTINUE
          RETURN
          END
```

Figure 7.16 Source code listing for subroutine UCLOSE.

7.3.3.2 FILIO.FOR

The routines STOFIL and RETFIL write and read the UnkelScope binary files. These routines should not be altered, or called directly, i.e., all calls to these should pass through the subroutines BINPUT and BOUTPU. The source listings for these routines are given as Figures 7.17 and 7.18.

7.3.3.3 FILIOA.FOR

The routines STOASC and RETASC write and read the UnkelScope ascii transfer files. The routines should not be altred or called directly, i.e., all calls to these should pass through the subroutine BINPUT and BOUTPU. The source listing up for these routines are given as Figures 7.19 and 7.20.

7.3.3.4 FILIOT.FOR

The routine STOTAB writes an UnkelScope ascii table file. Reading of ascii table format files is not supported so that RETTAB is a dummy routine that returns the NER=1 to denote an error. If a modification to STOTAB makes its output more easily read into the spreadsheet a database program you use, you can modify this subroutine appropriately.

The source listings for these routines are given as Figures 7.21 and 7.22.

```
      subroutine stofil(maxch,maxlen,ner)
      implicit character*1 (b)
      implicit integer*2 (i-n)
      COMMON/MDATAF/
     $ VOLT1(1024),HORZ1(1024),TIME1(1024),
     $ VOLT2(1024),HORZ2(1024),TIME2(1024),
     $ VOLT3(1024),HORZ3(1024),TIME3(1024)
     $ ,VOLT4(1024),HORZ4(1024),TIME4(1024)
      common/mdatan/nchd,npt1,npt2,npt3,npt4
      common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),bpv4la(20),bph4la(20),bpt4la(20)
      common/mdatah/bhead1(40),bhead2(40)
      common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
      integer*2 nd1,nd2
      CALL BSTUF(20,BHEAD1(1),'SCOPE Version 1.2 DB')
      call bstuf(20,bhead1(21),'                    ')
      ner=1
      WRITE(LSIO4,ERR=7001)BHEAD1,BHEAD2
      nd1=maxch
      nd2=maxlen
      WRITE(LSIO4,ERR=7001)nd1,nd2
      WRITE(LSIO4,ERR=7001)nd2
      WRITE(LSIO4,ERR=7001)(VOLT1(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(HORZ1(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(TIME1(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)BPV1LA,BPH1LA,BPT1LA
      if (maxch .le. 1)go to 780
      WRITE(LSIO4,ERR=7001)nd2
      WRITE(LSIO4,ERR=7001)(VOLT2(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(HORZ2(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(TIME2(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)BPV2LA,BPH2LA,BPT2LA
      if (maxch .le. 2)go to 780
      WRITE(LSIO4,ERR=7001)nd2
      WRITE(LSIO4,ERR=7001)(VOLT3(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(HORZ3(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(TIME3(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)BPV3LA,BPH3LA,BPT3LA
      if (maxch .le. 3)go to 780
      WRITE(LSIO4,ERR=7001)nd2
      WRITE(LSIO4,ERR=7001)(VOLT4(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(HORZ4(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)(TIME4(I),I=1,maxlen)
      WRITE(LSIO4,ERR=7001)BPV4LA,BPH4LA,BPT4LA
780   continue
      ner=0
      RETURN
7001  CONTINUE
7000  CONTINUE
      RETURN
      END
```

Figure 7.18 Source code listing for subroutine STOFIL.

```
      subroutine retfil(maxch,maxlen,ner)
      implicit character*1 (b),integer*2 (i-n)
      COMMON/MDATAF/
     $ VOLT1(1024),HORZ1(1024),TIME1(1024),
     $ VOLT2(1024),HORZ2(1024),TIME2(1024),
     $ VOLT3(1024),HORZ3(1024),TIME3(1024)
     $ ,VOLT4(1024),HORZ4(1024),TIME4(1024)
      common/mdatan/nchd,npt1,npt2,npt3,npt4
      common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),bpv4la(20),bph4la(20),bpt4la(20)
      common/mdatah/bhead1(40),bhead2(40)
      common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
      ner=1
      read(LSIO3,ERR=7001)BHEAD1,BHEAD2
      read(LSIO3,ERR=7001)nd1,nd2
      maxch=nd1
      maxlen=nd2
      read (LSIO3,ERR=7001)nd1
      npt1=nd1
      read (LSIO3,ERR=7001)(VOLT1(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(HORZ1(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(TIME1(I),I=1,maxlen)
      read (LSIO3,ERR=7001)BPV1LA,BPH1LA,BPT1LA
      if (maxch .le. 1)go to 780
      read (LSIO3,ERR=7001)nd1
      npt2=nd1
      read (LSIO3,ERR=7001)(VOLT2(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(HORZ2(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(TIME2(I),I=1,maxlen)
      read (LSIO3,ERR=7001)BPV2LA,BPH2LA,BPT2LA
      if (maxch .le. 2)go to 780
      read (LSIO3,ERR=7001)nd1
      npt3=nd1
      read (LSIO3,ERR=7001)(VOLT3(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(HORZ3(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(TIME3(I),I=1,maxlen)
      read (LSIO3,ERR=7001)BPV3LA,BPH3LA,BPT3LA
      if (maxch .le. 3)go to 780
      read (LSIO3,ERR=7001)nd1
      npt4=nd1
      read (LSIO3,ERR=7001)(VOLT4(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(HORZ4(I),I=1,maxlen)
      read (LSIO3,ERR=7001)(TIME4(I),I=1,maxlen)
      read (LSIO3,ERR=7001)BPV4LA,BPH4LA,BPT4LA
780   continue
      ner=0
      RETURN
7001  CONTINUE
7000  CONTINUE
      RETURN
      END
```

Figure 7.18 Source code listing for subroutine RETFIL.

```
      subroutine stoasc(maxch,maxlen,ner)
      implicit character*1 (B)
      implicit integer*2 (i-n)
      common/mdatan/nchd,npt1,npt2,npt3,npt4
      COMMON/MDATAF/
     $ VOLT1(1024),HORZ1(1024),TIME1(1024),
     $ VOLT2(1024),HORZ2(1024),TIME2(1024),
     $ VOLT3(1024),HORZ3(1024),TIME3(1024),
     $ VOLT4(1024),HORZ4(1024),TIME4(1024)
      common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),
     $ bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),
     $ bpv4la(20),bph4la(20),bpt4la(20)
      common/mdatah/bhead1(40),bhead2(40)
      common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
      DIMENSION BHEAD9(40)
      integer*2 nd1,nd2
      DATA BHEAD9/'S','C','O','P','E',' ','V','e','r',
     $ 's','i','o','n',' ','1','.','2',' ','D','A',20*' '/
      do 500 i=1,40
      bhead1(i)=bhead9(i)
500   continue
      ner=1
C HEADER LABEL
      WRITE(LSIO4,600,ERR=7001)BHEAD1,BHEAD2
600   format(80a1)
601   format(2i10)
602   format(258(4(1pe12.5,1x)/))
603   format(60a1)
      nd1=maxch
      nd2=maxlen
      WRITE(LSIO4,601,ERR=7001)nd1,nd2
      WRITE(LSIO4,601,ERR=7001)nd2
      WRITE(LSIO4,602,ERR=7001)(VOLT1(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(HORZ1(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(TIME1(I),I=1,maxlen)
      WRITE(LSIO4,603,ERR=7001)BPV1LA,BPH1LA,BPT1LA
      if (maxch .le. 1)go to 780
      WRITE(LSIO4,601,ERR=7001)nd2
      WRITE(LSIO4,602,ERR=7001)(VOLT2(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(HORZ2(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(TIME2(I),I=1,maxlen)
      WRITE(LSIO4,603,ERR=7001)BPV2LA,BPH1LA,BPT2LA
      if (maxch .le. 2)go to 780
      WRITE(LSIO4,601,ERR=7001)nd2
      WRITE(LSIO4,602,ERR=7001)(VOLT3(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(HORZ3(I),I=1,maxlen)
      WRITE(LSIO4,602,ERR=7001)(TIME3(I),I=1,maxlen)
      WRITE(LSIO4,603,ERR=7001)BPV3LA,BPH1LA,BPT3LA
```

Figure 7.19 Source code listing for subroutine STOASC.
Part 1 of 2.

```
          if (maxch .le. 3)go to 780
          WRITE(LSIO4,601,ERR=7001)nd2
          WRITE(LSIO4,602,ERR=7001)(VOLT4(I),I=1,maxlen)
          WRITE(LSIO4,602,ERR=7001)(HORZ4(I),I=1,maxlen)
          WRITE(LSIO4,602,ERR=7001)(TIME4(I),I=1,maxlen)
          WRITE(LSIO4,603,ERR=7001)BPV4LA,BPH1LA,BPT4LA
780       continue
          ner=0
          RETURN
7001      CONTINUE
7000      CONTINUE
          RETURN
          END
```

Figure 7.19 Source code listing for subroutine STOASC.
Part 2 of 2.

```
          subroutine retasc(maxch,maxlen,ner)
          implicit character*1 (B)
          implicit integer*2 (i-n)
          common/mdatan/nchd,npt1,npt2,npt3,npt4
          COMMON/MDATAF/
     $ VOLT1(1024),HORZ1(1024),TIME1(1024),
     $ VOLT2(1024),HORZ2(1024),TIME2(1024),
     $ VOLT3(1024),HORZ3(1024),TIME3(1024),
     $ VOLT4(1024),HORZ4(1024),TIME4(1024)
          common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),
     $ bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),
     $ bpv4la(20),bph4la(20),bpt4la(20)
          common/mdatah/
     $ bhead1(40),bhead2(40)
          common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
          integer*2 nd1,nd2
          ner=1
C HEADER LABEL
600       format(80a1)
601       format(2i10)
602       format(258(4(1pe12.5,1x)/))
603       format(60a1)
          read(LSIO3,600,ERR=7001)BHEAD1,BHEAD2
          read(LSIO3,601,ERR=7001)nd1,nd2
          maxch=nd1
          maxlen=nd2
          read (LSIO3,601,ERR=7001)nd1
```

Figure 7.20 Source code listing for subroutine RETASC.
Part 1 of 2.

```
      npt1=nd1
      read (LSI03,602,ERR=7001)(VOLT1(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(HORZ1(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(TIME1(I),I=1,maxlen)
      read (LSI03,603,ERR=7001)BPV1LA,BPH1LA,BPT1LA
      if (maxch .le. 1)go to 780
      read (LSI03,601,ERR=7001)nd1
      npt2=nd1
      read (LSI03,602,ERR=7001)(VOLT2(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(HORZ2(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(TIME2(I),I=1,maxlen)
      read (LSI03,603,ERR=7001)BPV2LA,BPH2LA,BPT2LA
      if (maxch .le. 2)go to 780
      read (LSI03,601,ERR=7001)nd1
      npt3=nd1
      read (LSI03,602,ERR=7001)(VOLT3(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(HORZ3(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(TIME3(I),I=1,maxlen)
      read (LSI03,603,ERR=7001)BPV3LA,BPH3LA,BPT3LA
      if (maxch .le. 3)go to 780
      read (LSI03,601,ERR=7001)nd1
      npt4=nd1
      read (LSI03,602,ERR=7001)(VOLT4(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(HORZ4(I),I=1,maxlen)
      read (LSI03,602,ERR=7001)(TIME4(I),I=1,maxlen)
      read (LSI03,603,ERR=7001)BPV4LA,BPH4LA,BPT4LA
780   continue
      ner=0
      RETURN
7001  CONTINUE
7000  CONTINUE
      RETURN
      END
```

Figure 7.20 Source code listing for subroutine RETASC.
Part 2 of 2.

```
          subroutine stotab(maxch,maxlen,ner)
          implicit character*1 (b)
          implicit integer*2 (i-n)
          COMMON/MDATAF/
     $ VOLT1(1024),HORZ1(1024),TIME1(1024),
     $ VOLT2(1024),HORZ2(1024),TIME2(1024),
     $ VOLT3(1024),HORZ3(1024),TIME3(1024)
     $ ,VOLT4(1024),HORZ4(1024),TIME4(1024)
          common/mdatan/nchd,npt1,npt2,npt3,npt4
          common/mdatal/
     $ bpv1la(20),bph1la(20),bpt1la(20),
     $ bpv2la(20),bph2la(20),bpt2la(20),
     $ bpv3la(20),bph3la(20),bpt3la(20),
     $ bpv4la(20),bph4la(20),bpt4la(20)
          common/mdatah/bhead1(40),bhead2(40)
          common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
          integer*2 nd1,nd2
          CALL BSTUF(20,BHEAD1( 1),'SCOPE Version 1.2 DT')
          call bstuf(20,bhead1(21),'                    ')
          ner=1
c call output routine for first two
          call outtab(maxch,maxlen,bhead2,
     $           npt1,bpt1la,bpv1la,bph1la,time1,volt1,horz1,
     $           npt2,bpt2la,bpv2la,bph2la,time2,volt2,horz2)
c call output routine for second two
          mm2=maxch-2
          call outtab(mm2  ,maxlen,bhead2,
     $           npt3,bpt3la,bpv3la,bph3la,time3,volt3,horz3,
     $           npt4,bpt4la,bpv4la,bph4la,time4,volt4,horz4)
          return
          end
          subroutine outtab(maxch,maxlen,blabel,
     $    npt1,bt1,bv1,bh1,t1,v1,h1,npt2,bt2,bv2,bh2,t2,v2,h2)
          implicit character*1 (b)
          implicit integer*2 (i-n)
          dimension blabel(40),bt1(20),bv1(20),bh1(20),
     $    bt2(20),bv2(20),bh2(20),
     $    t1(1),v1(1),h1(1),t2(1),v2(1),h2(1)
          common/munits/luset,ludis,lsio1,lsio2,lsio3,lsio4
c if maxch is less than or equal to 0
          if(maxch .le. 0)return
          write(lsio4,6000)(blabel(i),i=1,40)
6000      format('"',40a1'"')
c if maxch is 1 then use single set format
          if(maxch .gt. 1)go to 7000
c single set format
          write(lsio4,601)(bt1(i),i= 1,10)
     $                   ,(bv1(i),i= 1,10)
     $                   ,(bh1(i),i= 1,10)
```

Figure 7.21 Source code listing for subroutine STOTAB.
(Part 1 of 2.)

```
        write(lsio4,601)(bt1(i),i=11,20)
     $                  ,(bv1(i),i=11,20)
     $                  ,(bh1(i),i=11,20)
601     format('"#"',3('"',10a1,'"'))
        nfirst=1
        nlast=maxlen
        nint=1
        do 50 i=nfirst,nlast,nint
        write(lsio4,602)i,t1(i),v1(i),h1(i)
50      continue
602     format(i5,3(1x,f10.5))
        return
7000    continue
c pair format
        write(lsio4,611)(bt1(i),i= 1,10)
     $                  ,(bv1(i),i= 1,10)
     $                  ,(bh1(i),i= 1,10)
     $                  ,(bv2(i),i= 1,10)
     $                  ,(bv2(i),i= 1,10)
     $                  ,(bh2(i),i= 1,10)
        write(lsio4,611)(bt1(i),i=11,20)
     $                  ,(bv1(i),i=11,20)
     $                  ,(bh1(i),i=11,20)
     $                  ,(bv2(i),i=11,20)
     $                  ,(bv2(i),i=11,20)
     $                  ,(bh2(i),i=11,20)
611     format('"#"',3('"',10a1,'"'),'"#"',3('"',10a1,'"'))
        nfirst=1
        nlast=maxlen
        nint=1
        do 51 i=nfirst,nlast,nint
        write(lsio4,612)i,t1(i),v1(i),h1(i),i,t2(i),v2(i),h2(i)
51      continue
612     format(i5,3(1x,f10.5),i5,3(1x,f10.5))
        RETURN
        END
```

Figure 7.21 Source code listing for subroutine STOTAB.
(Part 2 of 2.)

```
        subroutine rettab(maxch,maxlen,ner)
        implicit integer*2 (i-n)
        ner=1
c print message that cannot read table data this way
        write(*,60)
60      format(' Cannot read data from table format')
        RETURN
        END
```

Figure 7.22 Source code listing for subroutine RETTAB.

7.4 Manipulating UnkelScope Data Files Within the Basic Environment

To allow the most flexibility for the BASIC(A) programmer, the listing is provided for a 'template' program that allows offline manipulation of data from files. Unfortunately BASIC cannot directly read the UnkelScope binary files. Thus the first step in processing data with BASIC is to run the program BINASC (see Section 7.2) to create an ascii transfer file from the binary file written by the BASIC program. If the output file is to be used by the UnkelScope program, the program ASCBIN (see Section 7.2) must be used to convert the ascii transfer file to a binary file that can be read by the UnkelScope program.

This section assumes the reader has a working knowledge of the BASIC language and the mechanics of creating and modifying programs. The program provided as an example takes a file with sampled data of the instantaneous voltage and current to determine the instantaneous power dissipated in a device and the total energy used from the start of the scan. The voltage is stored in Vertical Trace 1 and the current is stored in Vertical Trace 2. Nothing is stored in Vertical Traces 3 or 4. Figures 7.23a and 7.23b show the setup configurations and representative data for this example. The instantaneous power is stored as Horizontal Trace 1 and the total energy is stored as Horizontal Trace 2. This approach (as opposed to using Vertical Trace 3 and Vertical Trace 4) saves space on the disk.

Figure 7.24 shows a flow chart of this template program; the code w ith comments is shown as Figure 7.25. The code is divided roughly into 5 sections:

Statements	5	to	70:	declare actions;
Statements	100	to	103:	get the file names;
Statements	200	to	704:	read the input file;
Statements	1000	to	1014:	process the data;
Statements	9000	to	9124:	write the output file.

The basic statements from 1000 to 8999 are available for the actual code you wish to write. The rest of the material should not be modified, although in some cases there may be no need to read an input file so that secti on might be removed.

Note that if the largest trace number with valid data is altered by the processing step, the variable MAXCH must be altered. Also if the longest length of the arrays is altered, then the variable MAXLEN must be adjusted appropriately.

The screen dialog to process the data using the basic program PROCES is shown as Figure 7.26a. The setup configuration and plotted results of the processing are shown in Figure 7.26b and 7.26c.

```
-------- Vertical Trace 1 --------       -------- Additional Vertical Traces --------
Source [Analog  0]  A/D Range [+/- 10]   Tr Input                               A/D
Label  :Voltage (volts)                  #  Chan  Label:                        Range
Span   [ 10   v full scale]              3 [none ]
Range  [-5.00E+00 to  5.00E+00]          4 [none ]

-------- Vertical Trace 2 --------
Source [Analog  1]  A/D Range [+/- 10]
Label  :Current (amps)
Span   [ 10   v full scale]
Range  [  .00E+00 to  1.00E+01]          -------- Triggering --------
                                         Mode  [Singl Sweep] Source [Keyboard ]
-------- Horizontal Trace --------
Source [Time      ]
Label  :Time(Seconds)                    -------- Processing --------
Span   [ 1    s full scale]              Type [none                       ]
Range  [  .00E+00 to  1.00E+00]

-------- Sampling --------
Sample Rate [  1 ms        1  khz]
[1024] Samples  {Scan Time 1.02E+00 s}
{Delayed Plot  } {Processing Inactive}
```

Figure 7.23a Setup configuration to take data for example program PROCES.

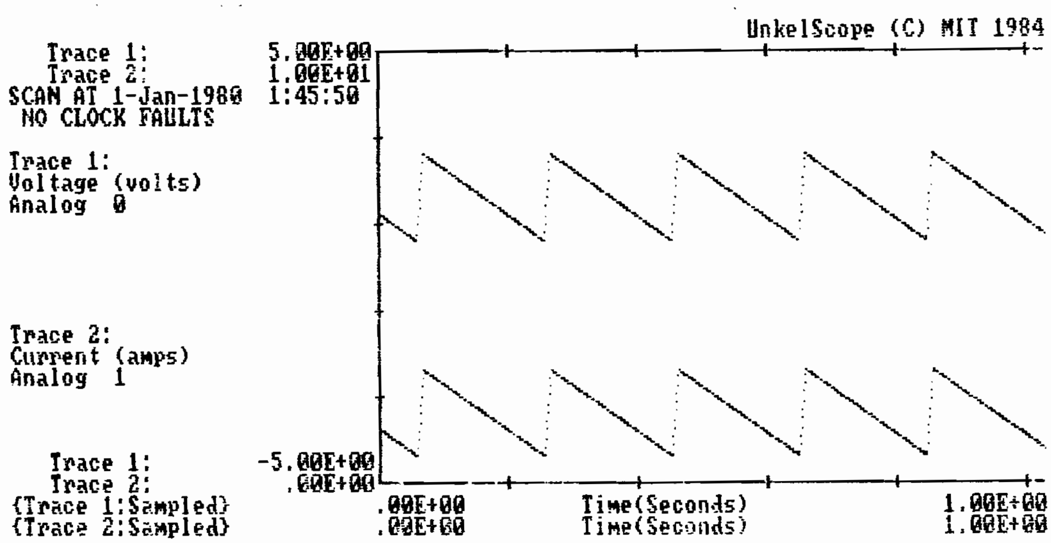


Figure 7.23b Representative data to be processed by program PROCES.

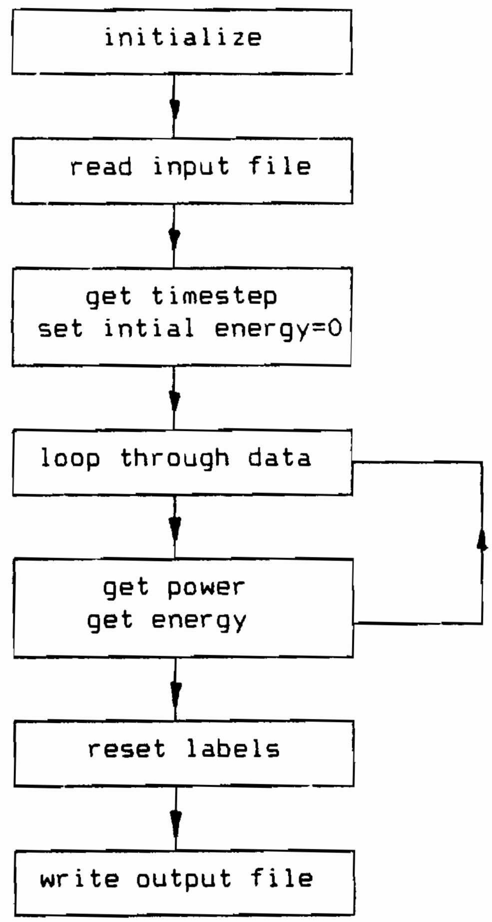


Figure 7.24 Flowchart for program PROCES.

```
5 REM first make variable type assignments
6 REM   fortran type except "b" which are strings
10 DEFINT I-N
20 DEFSNG A,C-H,O-Z
30 DEFSTR B
35 REM now dimension the data arrays
40 DIM VERT1(1024),HORZ1(1024),TIME1(1024)
50 DIM VERT2(1024),HORZ2(1024),TIME2(1024)
60 DIM VERT3(1024),HORZ3(1024),TIME3(1024)
70 DIM VERT4(1024),HORZ4(1024),TIME4(1024)
100 REM now prompt for and read the input file name
101 INPUT;" Type Input File Name (10 characters or less):",BINPUT$
102 PRINT " "
103 INPUT;"Type Output File Name (10 characters or less):",BOUTPU$
200 REM now starts the program "proces"
201 REM must open and read the file
202 OPEN BINPUT$ FOR INPUT AS #1
203 REM read labels
204 BHEAD1$=INPUT$(40,#1)
205 BHEAD2$=INPUT$(40,#1)
206 REM read maximum channels and number of data points each array
207 INPUT #1,MGARB,MAXCH,MAXLEN
209 REM  read trace1 if there is one
211 IF (MAXCH < 1) THEN GOTO 297
212 INPUT #1,NPT1
213 FOR I=1 TO MAXLEN STEP 4
214 INPUT #1,VERT1(I),VERT1(I+1),VERT1(I+2),VERT1(I+3)
215 NEXT
216 INPUT #1,M
217 FOR I=1 TO MAXLEN STEP 4
218 INPUT #1,HORZ1(I),HORZ1(I+1),HORZ1(I+2),HORZ1(I+3)
219 NEXT
220 INPUT #1,M
221 FOR I=1 TO MAXLEN STEP 4
222 INPUT #1,TIME1(I),TIME1(I+1),TIME1(I+2),TIME1(I+3)
223 NEXT
224 INPUT #1,M
225 REM now the label line is read and split into correct parts
226 INPUT #1,BLONG$
227 BPV1LA$=MID$(BLONG$,1,20)
228 BPH1LA$=MID$(BLONG$,21,20)
229 BPT1LA$=MID$(BLONG$,41,20)
230 REM
231 REM read trace2 if there is one
232 REM
233 IF (MAXCH < 2) THEN GOTO 297
234 INPUT #1,NPT2
235 FOR I=1 TO MAXLEN STEP 4
236 INPUT #1,VERT2(I),VERT2(I+1),VERT2(I+2),VERT2(I+3)
237 NEXT
```

Figure 7.25 Source code listing for BASIC program PROCES.
(Page 1 of 5.)

```
238 INPUT #1,M
239 FOR I=1 TO MAXLEN STEP 4
240 INPUT #1,HORZ2(I),HORZ2(I+1),HORZ2(I+2),HORZ2(I+3)
241 NEXT
242 INPUT #1,M
243 FOR I=1 TO MAXLEN STEP 4
244 INPUT #1,TIME2(I),TIME2(I+1),TIME2(I+2),TIME2(I+3)
245 NEXT
246 INPUT #1,M
247 REM now the label line is read and split into correct parts
248 INPUT #1,BLONG$
249 BPV2LA$=MID$(BLONG$,1,20)
250 BPH2LA$=MID$(BLONG$,21,20)
251 BPT2LA$=MID$(BLONG$,41,20)
252 REM
253 REM read trace3 if there is one
254 REM
255 IF (MAXCH < 3) THEN GOTO 297
256 INPUT #1,NPT3
257 FOR I=1 TO MAXLEN STEP 4
258 INPUT #1,VERT3(I),VERT3(I+1),VERT3(I+2),VERT3(I+3)
259 NEXT
260 INPUT #1,M
261 FOR I=1 TO MAXLEN STEP 4
262 INPUT #1,HORZ3(I),HORZ3(I+1),HORZ3(I+2),HORZ3(I+3)
263 NEXT
264 INPUT #1,M
265 FOR I=1 TO MAXLEN STEP 4
266 INPUT #1,TIME3(I),TIME3(I+1),TIME3(I+2),TIME3(I+3)
267 NEXT
268 INPUT #1,M
269 REM now the label line is read and split into correct parts
270 INPUT #1,BLONG$
271 BPV3LA$=MID$(BLONG$,1,20)
272 BPH3LA$=MID$(BLONG$,21,20)
273 BPT3LA$=MID$(BLONG$,41,20)
274 REM
275 REM read trace4 if there is one
276 REM
277 IF (MAXCH < 4) THEN GOTO 297
278 INPUT #1,NPT4
279 FOR I=1 TO MAXLEN STEP 4
280 INPUT #1,VERT4(I),VERT4(I+1),VERT4(I+2),VERT4(I+3)
281 NEXT
282 INPUT #1,M
283 FOR I=1 TO MAXLEN STEP 4
284 INPUT #1,HORZ4(I),HORZ4(I+1),HORZ4(I+2),HORZ4(I+3)
```

Figure 7.25 Source code listing for BASIC program PROCES.
(Page 2 of 5.)

```
285 NEXT
286 INPUT #1,M
287 FOR I=1 TO MAXLEN STEP 4
288 INPUT #1,TIME4(I),TIME4(I+1),TIME4(I+2),TIME4(I+3)
289 NEXT
290 INPUT #1,M
291 REM now the label line is read and split into correct parts
292 INPUT #1,BLONG$
293 BPV4LA$=MID$(BLONG$,1,20)
294 BPH4LA$=MID$(BLONG$,21,20)
295 BPT4LA$=MID$(BLONG$,41,20)
296 REM
297 REM file has been read print message about number of traces and label
298 PRINT BHEAD2$
299 PRINT "file has ",MAXCH," channel(s) of data"
300 REM now close the file
301 CLOSE #1
704 REM this ends the section to read the input file
1000 REM now starts the processing portions
1001 REM determine the timestep assuming points evenly spaced
1002 DTIME=TIME1(2)-TIME1(1)
1003 REM set initial energy equal to zero
1004 ENERGY=0!
1005 REM now the loop to do the processing
1006 FOR I=1 TO NPT1 STEP 1
1007 POWER=VERT1(I)*VERT2(I)  'compute the power
1008 ENERGY=ENERGY+POWER*DTIME 'compute the energy
1009 HORZ1(I)=POWER/1000!
1010 HORZ2(I)=ENERGY
1011 NEXT
1012 REM now get the labels set properly
1013 BPH1LA$="   Power (kwatts)   "
1014 BPH2LA$="  Energy (Joules)   "
9000 REM
9001 REM write trace1 if there is one
9002 REM
9003 REM open the output file
9004 OPEN BOUTPU$ FOR OUTPUT AS #1
9005 PRINT #1,USING"\                                            \";BHEAD1$;
9006 PRINT #1,USING"\                                            \";BHEAD2$
9007 PRINT #1,USING"      ####      ####";MAXCH,MAXLEN
9008 IF (MAXCH < 1) THEN GOTO 9075
9009 PRINT #1,USING"      ####";MAXLEN
9010 FOR I=1 TO MAXLEN STEP 4
9011 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";VERT1(I),VERT1(I+1),VERT1(I+2),VERT1(I+3)
9012 NEXT
9013 PRINT #1," "
9014 FOR I=1 TO MAXLEN STEP 4
9015 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";HORZ1(I),HORZ1(I+1),HORZ1(I+2),HORZ1(I+3)
```

Figure 7.25 Source code listing for BASIC program PROCES.
Part 3 of 5.

```
9016 NEXT
9017 PRINT #1," "
9018 FOR I=1 TO MAXLEN STEP 4
9019 PRINT#1, USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";TIME1(I),TIME1(I+1),TIME1(I+2),TIME1(I+3)
9020 NEXT
9021 PRINT #1," "
9022 REM now the labels
9023 PRINT #1,USING"\                          \";BPV1LA$,BPH1LA$,BPT1LA$
9024 REM now do trace 2 if needed
9025 IF (MAXCH < 2) THEN GOTO 9075
9026 PRINT #1,USING"        ####";MAXLEN
9027 FOR I=1 TO MAXLEN STEP 4
9028 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";VERT2(I),VERT2(I+1),VERT2(I+2),VERT2(I+3)
9029 NEXT
9030 PRINT #1," "
9031 FOR I=1 TO MAXLEN STEP 4
9032 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";HORZ2(I),HORZ2(I+1),HORZ2(I+2),HORZ2(I+3)
9033 NEXT
9034 PRINT #1," "
9035 FOR I=1 TO MAXLEN STEP 4
9036 PRINT#1, USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";TIME2(I),TIME2(I+1),TIME2(I+2),TIME2(I+3)
9037 NEXT
9038 PRINT #1," "
9039 REM now the labels
9040 PRINT #1,USING"\                          \";BPV2LA$,BPH2LA$,BPT2LA$
9041 REM now do trace 2 if needed
9042 IF (MAXCH < 3) THEN GOTO 9075
9043 PRINT #1,USING"        ####";MAXLEN
9044 FOR I=1 TO MAXLEN STEP 4
9045 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";VERT3(I),VERT3(I+1),VERT3(I+2),VERT3(I+3)
9046 NEXT
9047 PRINT #1," "
9048 FOR I=1 TO MAXLEN STEP 4
9049 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";HORZ3(I),HORZ3(I+1),HORZ3(I+2),HORZ3(I+3)
9050 NEXT
9051 PRINT #1," "
9052 FOR I=1 TO MAXLEN STEP 4
9053 PRINT#1, USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";TIME3(I),TIME3(I+1),TIME3(I+2),TIME3(I+3)
9054 NEXT
9055 PRINT #1," "
9056 REM now the labels
9057 PRINT #1,USING"\                          \";BPV3LA$,BPH3LA$,BPT3LA$
9058 REM now do trace 2 if needed
```

Figure 7.25 Source code listing for BASIC program PROCES.
Part 4 of 5.

```
9059 IF (MAXCH < 4) THEN GOTO 9075
9060 PRINT #1,USING"      ####";MAXLEN
9061 FOR I=1 TO MAXLEN STEP 4
9062 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";VERT4(I),VERT4(I+1),VERT4(I+2),VERT4(I+3)
9063 NEXT
9064 PRINT #1," "
9065 FOR I=1 TO MAXLEN STEP 4
9066 PRINT #1,USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
     ##.#####^^^^";HORZ4(I),HORZ4(I+1),HORZ4(I+2),HORZ4(I+3)
9067 NEXT
9068 PRINT #1," "
9069 FOR I=1 TO MAXLEN STEP 4
9070 PRINT#1, USING"##.#####^^^^ ##.#####^^^^ ##.#####^^^^
##.#####^^^^";TIME4(I),TIME4(I+1),TIME4(I+2),TIME4(I+3)
9071 NEXT
9072 PRINT #1," "
9073 REM now the labels
9074 PRINT #1,USING"\                  \";BPV4LA$,BPH4LA$,BPT4LA$
9075 CLOSE #1
9124 REM this is the end of the program
```

Figure 7.25 Source code listing for BASIC program PROCES.
Part 5 of 5.

```
C>binasc
 input data file name?ramps.dat
output data file name?ramps.asc
C>basica
   OK
   LOAD"proces.lst",R
    Type Input File Name (10 characters or less):ramps.asc
   Type Output File Name (10 characters or less):rampm.asc
C>ascbin
 input data file name?rampm.asc
output data file name?rampm.pow
```

Figure 7.26a Screen dialog to use PROCES to process data from UnkelScope.

```
---------- Vertical Trace 1 ----------    ---------- Additional Vertical Traces ----------
Source [File Hor1]File Name:ramps.pow    Tr Input                              A/D
Label  :   Power (kwatts)                #  Chan  Label:                       Range
Span   [50    mv full scale]             3 [none ]
Range  [-2.50E-02 to  2.50E-02]          4 [none ]

---------- Vertical Trace 2 ----------
Source [File Hor2]File Name:ramps.pow
Label  :  Energy(kJoules)
Span   [10    mv full scale]
Range  [   .00E+00 to  1.00E-02]         ---------- Triggering ----------
                                         Mode  [Singl Sweep] Source [Keyboard ]
---------- Horizontal Trace ----------
Source [Time      ]
Label  :Time(Seconds)                    ---------- Processing ----------
Span   [ 1    s full scale]              Type [none                      ]
Range  [   .00E+00 to  1.00E+00]

---------- Sampling ----------
Sample Rate [  1 ms         1  khz]
```

Figure 7.26b Setup configuration to display the results of the program PROCES.

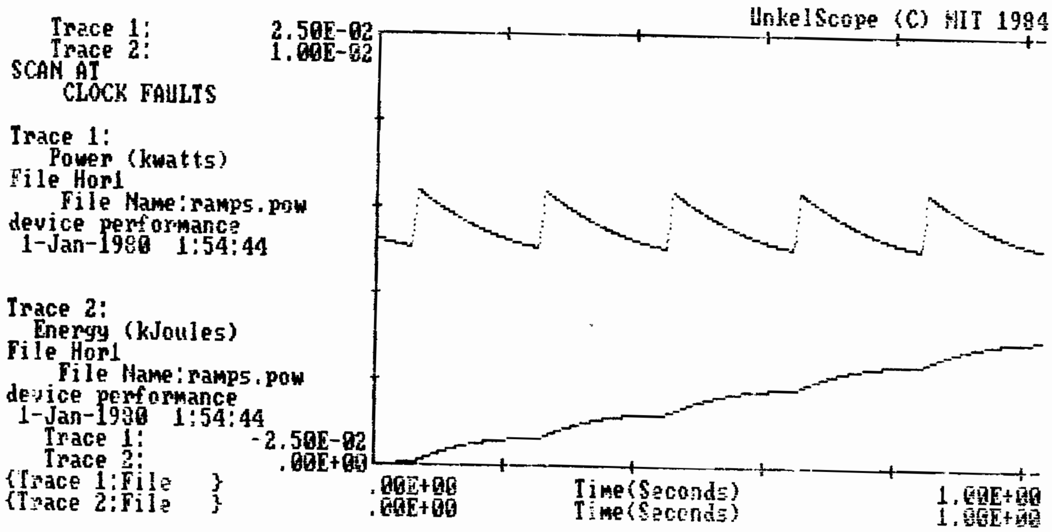


Figure 7.26c Plotted results of processing the data of Figure 7.23b with the program PROCES.

8.0 UnkelScope in the Context of Computer-Based Laboratory Systems

Before introducing the additional features of UnkelScope, it is important to review the capabilities of the software system in the general context of engineering and scientific research. A schematic view is given in Figure 8.1 using the data flow path as the main orientation, but including most of the features that are important in an investigation that has a significant component of laboratory work. The view is meant to be general and does not need to include a computer or any digital based system. In fact, the overall goal is to use the computer to increase the effectiveness in accomplishing goals within the context of this general approach.

The main path of information from the experiment starts from a transducer, then passes through a signal conditioner to the recording instrument. Some information about time and perhaps a means of synchronizing data taking with events of the apparatus are usually required. This recorded data may be displayed, and then processed and redisplayed. In some cases the data is used to control the experiment either in a control loop or to adjust the parameter settings. An actuator system is used to accomplish this.

The "archive" is used to store information and may help coordinate the experimental activities with the analytical and documentation processes. The final activities are of design and redesign of equipment based upon the results and the supervisory task of coordinating and setting up the process.

Within this context, the portion of UnkelScope described in Part 1 (Level 1 of the software package) concentrates on the activities within the dashed line. The first goal of the software is to provide an easy and well documented way to acquire, display and store data from the experiment using the computer system. With no further enhancement the computer can then be used as a substitute for analog based equipment, subject to the constraints on speed and resolution of the computer system. While this first step is necessary, and in itself significant, the major advantages of the computer system lie in what can be done because the data is stored by the computer system and because the computer is available to organize and document the entire process. Specifically the 'enhancements' include:

a) data processing (and re-display);

b) comparison of theoretical and experimental results;

c) control of experiments in real-time;

d) organization, selection and documentation of the procedure and conditions themselves;

e) graphical editing of data files;

f) procedures (macros); and,

g) general examination of data files.

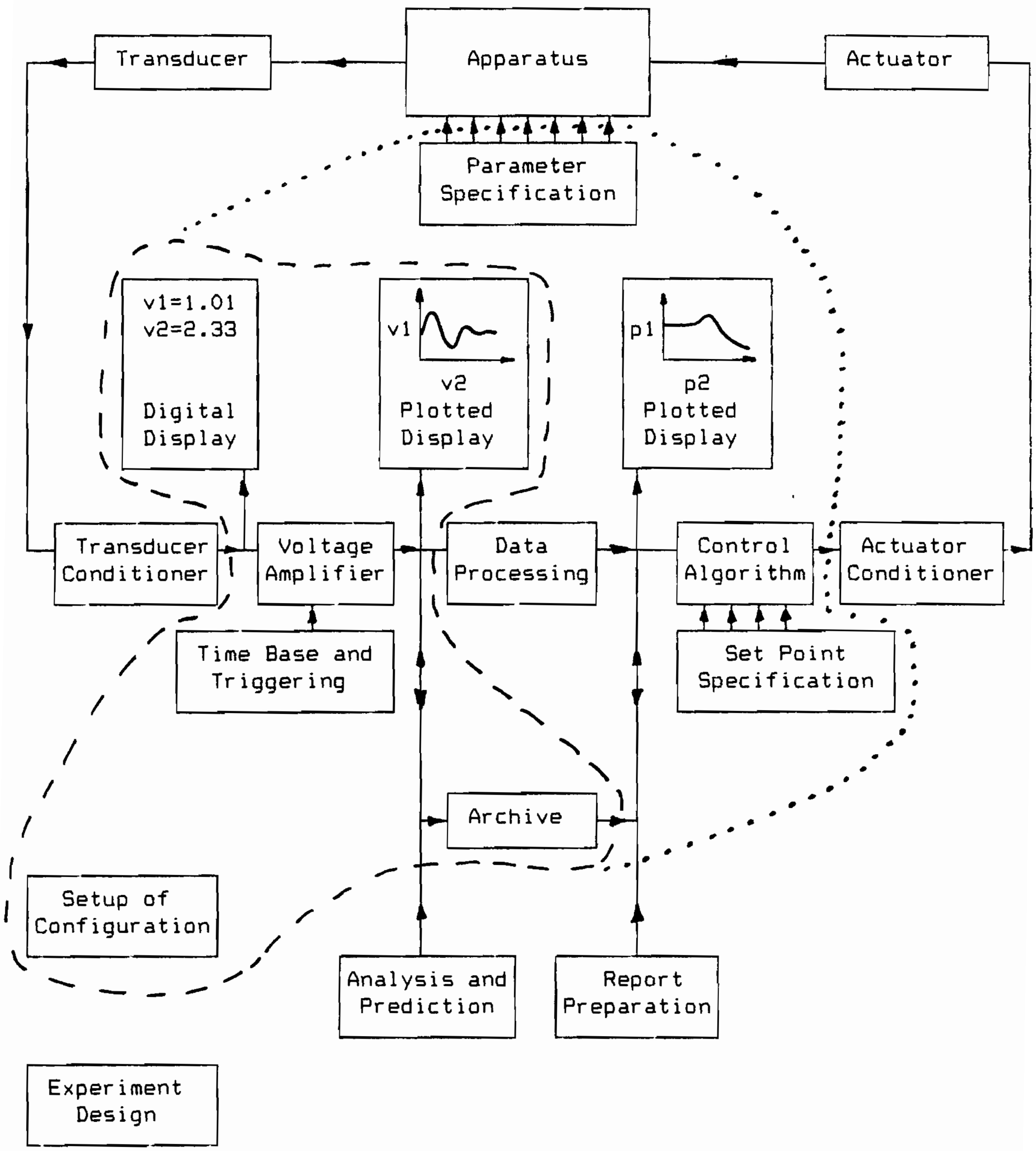


Figure 8.1 Schematic of general measurement process. The dashed line indicates the functions described in Part 1 of the Users Guide. The dotted line indicates the functions described in Part 2 and Part 3.

The the above activities can be done without the computer, however they can be done more effectively using the computer as a tool. For example, an analog based device such as an oscilloscope can create a hardcopy record of a transient trace and this may under some circumstances be sufficient. But when it is necessary to convert the voltage reading to pressure, for example the inconvenience of the analog record (photograph) is apparent.

The natural extension of the software would be to have all data processing options proceeding online as a modification to the flow of data. The software at approaches this as best as the hardware can support. The remainder of this section describes data processing in the context of the UnkelScope program and divides the activities into "online," "out-of-line" and "offline" activities. Online activities are those that are performed in the main flow of data through the program; in the Level 2 software the only online processing is linear conversion; control functions are also online. Out-of-line processing covers functions within the program; these processing functions, termed Utilities, take input data from previously stored disk files, process the data and then write the revised data to a disk file that can be accessed by UnkelScope. Like "out-of-line" processing, offline processing starts and finishes with data stored in disk files. However, in the offline programming environments the processing is done in a separate program rather than as a "Utility" of the program; Section 7 has discussed offline processing. Section 8.1 describes, in general terms, the Utilities i.e. out-of-line processing. Section 8.2 discusses control related functions. Section 8.3 discusses the online conversion capability.

8.1 Data Processing Utilities

UnkelScope provides 5 types of data processing activities as out-of-line Utilities:

a) Linear Calibration and out-of-line Conversion of Data (Section 9);

b) Plotting of Tables of Numbers Entered through the Keyboard (Section 10);

c) Statistical Analysis of Data (Section 11);

d) Integration, Differentiation and Other Functions Applied to Data (Section 12); and,

e) Graphical Editing of Data (Section 15.)

To help combine these processing capabilities into easy to execute "procedures" (macros), UnkelScope has "Make Procedure" and "Get Procedure" capabilities (Section 16.)

8.2 Control

Control can range from a closed-loop control system to a system responding to critical conditions by shutting down a particular device. In the former case a digital-to-analog converter provides the output signal. In the latter case the digital output lines effect the on/off control. UnkelScope includes a special processing option block that allows for relatively simple implementation of a closed loop controller and control of 4 digital output lines. Another common experimental control/data sampling configuration is a "traversing" situation where a parameter is altered (stepped) then data is taken and the process repeated. The traversing can be accomplished with a stepper motor driven by digital output lines or with a device that is controlled by the digital-to-analog output. The user interfaces for these functions are described in detail in Section 13.

8.3 Online Processing

The program allows online conversion of measured data according to a linear conversion equation. This feature is described in Section 14.

Passing Data from Utility to Utility without writing Disk Files

Under some circumstances you may not wish to see the intermediate results of calculations done by the Utilities. If this is the case, you can have UnkelScope pass the information from step to step provided that you:

a) do not leave the Utilities in between steps, and

b) you store the last step (and some of the others along the way.)

To use this feature specify a file name of = (the character =) as the output of one step and as the input of the next step.

The reason that you cannot exit the Utility section is because UnkelScope shares the storage for the Utilities with some of the storage for sampled data. If you try to exit the Utilities when the last output file was specified as "=", UnkelScope will give you a chance to go back and save this data (simply execute one of the utilities with no changes made except for specifying an output file name.) If you do not save the data at this time, the data will be destroyed and you will not be able to save it later.

9.0 Static Calibration of Transducers and Conversion of Data

An analog-to-digital converter measures the voltage of the input signal. In many cases this input signal has been created by a transducer that "measures" the physical parameter of interest. A pressure transducer that converts an input pressure to a voltage, a thermocouple and a hot-wire velocity probe are examples of transducers commonly used. The transduction to voltage is made because it is generally easier to detect, record and process voltage levels. However, the parameter of real interest is not the voltage and it is necessary to convert the measured voltage signal in order to evaluate the results - for example to compare experimental results to analytical results. Under many conditions this is done by calibrating the transducer in a series of static measurements.

The calibration of the transducer is made either:

1) using an accurate standard gage or transducer; or,

2) using a simple "primary" type measurement.

An example of the first would be to buy and use a pressure gage calibrated at the National Bureau of Standards - or at least one "known" to be accurate. An example of the second would be to use a manometer and a ruler as an accurate measure of pressure. In either case, the static calibration process consists of creating a condition with a certain value of the physical parameter - e.g. pressure - and recording the value of the "standard" and the value of the voltage from the transducer. This table of values can be used in a variety of ways to later convert the measured output of the transducer to a value of the quantity measured.

The conversion relation can be expressed in many formats. One commonly used approach is to choose an algebraic form for the input/output relationship and use the calibration data to determine the numerical values for the "parameters." Then the equation is used to algebraically convert the values. To obtain an accurate fit with a minimum of constants it is important to select the functional form of the fitting curve to correspond to the physical relationship governing the transducer. For example a hot-film anemometer is expected (from a theoretical analysis) to behave according to the relationship

$$V_2 = A + B \sqrt{v}$$

where V is the output voltage, v is the velocity being measured and A, B are constants that depend on the particular device. It would be unprofitable to try to fit calibration information from this device with a linear equation of the form, $V = mv + b$ or even with a series of power terms. Thus, care must be exercised in the curve fitting process. Fortunately many transducers are close to linear devices, especially if the operating range is not large.

The error in the curvefit must be presented in the most graphic and accurate fashion. Often the statistical quantity to describe the fit is not a good representation of the quality of the fit. A plot of the calibration curve compared to the calibration data is usually the best way to evaluate the fit and is at least an extremely useful supplement to the statistical quantity representing the error.

Realizing that the linear transducer is at best an approximation that applies in a limited range, the next section reviews the technique of linear regression as a means to determine the calibration function for a transducer.

9.1 Linear, Least Squares Curvefitting of Data

There are many numerical techniques for fitting curves depending on the assumed form of the curve and the parameter defining the "goodness" of the fit. The simplest technique is that of fitting a straight line using the least squares "goodness" parameter. Consider a sequence of data pairs (x_i, y_i) where y_i is considered the dependent variable and x_i the independent variable. As indicated in Figure 9.1, the least squares technique chooses the best line by minimizing the sum of the squares of the vertical distances between each particular data point and the straight line fit. The error between the curve and one data point is expressed as

$$e_i = (y_i - y_{predicted}(x_i))^2$$

where

$$y_{predicted} = m\ x_i + b$$

and m and b are the (to be determined) curvefit slope and intercept respectively. The best fit is determined by finding the minimum of the sum of these squares with respect to both m and b. For a system of n data points the resulting equations are

$$m = \frac{(\Sigma x_i)(\Sigma y_i) - n\Sigma(x_i y_i)}{(\Sigma x_i)(\Sigma x_i) - n\Sigma(x_i x_i)}$$

$$b = (\Sigma y_i - m\ \Sigma x_i)/n$$

where the summation runs from 1 to n.

There are several techniques to numerically represent the overall "goodness" of the fit. Some common ones are the standard deviation and the probable error. As mentioned earlier a graphical comparison of the curvefit data and curvefit is an important part of evaluating the goodness of the fit.

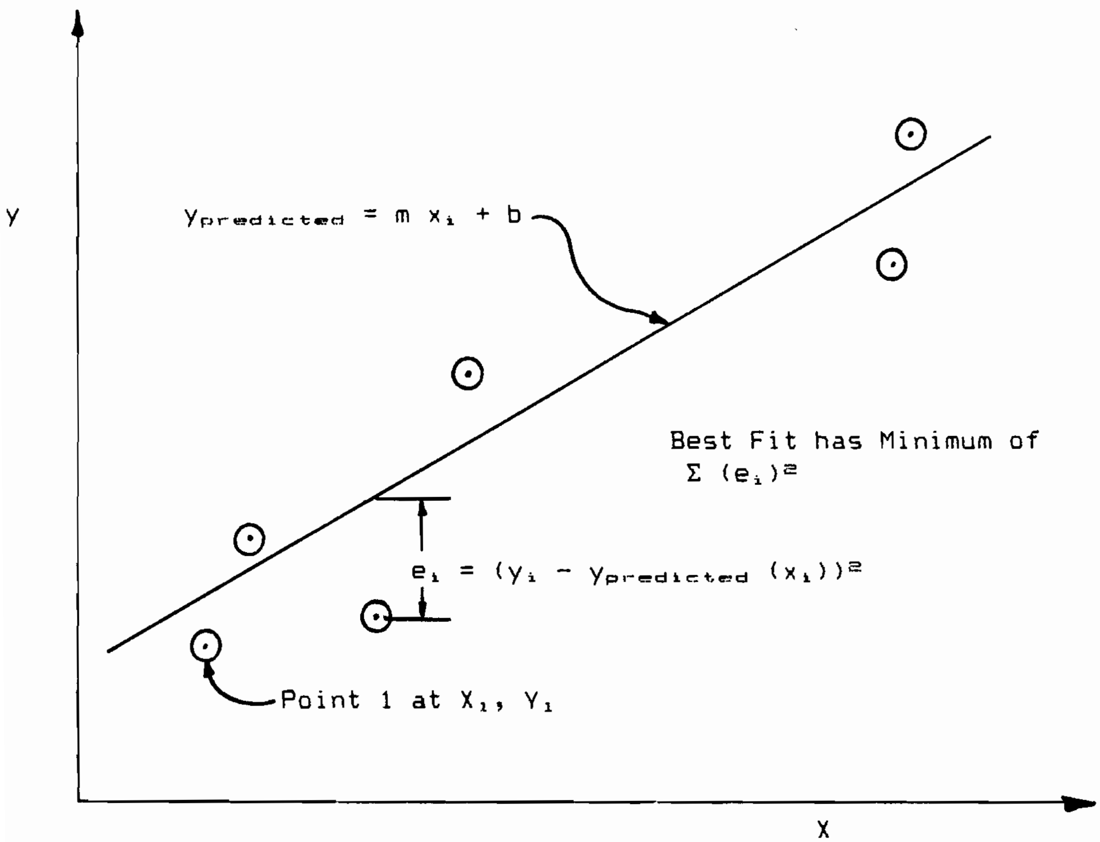


Figure 9.1 Schematic of linear least squares curvefitting technique.

9.2 Curvefitting Other Forms

Polynomial fitting is an extension of techniques described above and can be handled by extending least-squares fitting technique in straightforward fashion. There are many other techniques of curvefitting including "spline" fits and the reader is referred elsewhere for a discussion of these techniques.

There are also other equations that can be put in a form such that the linear least squares techniques can be applied. Two examples are

$$y = Ax^{n} \qquad (1)$$

and

$$y = A\,e^{mx} \qquad (2)$$

In the case of Equation 1, taking the natural logarithm of both sides yields

$$\ln y = \ln A + n \ln x$$

such that a linear least squares fit of the data pair ($\ln y_i$, $\ln x_i$) yields ln A as the intercept and n as the slope. In a similar fashion, taking the common logarithm of Equation 2 results in

$$\ln y = \ln A + m\,x$$

and a linear least squares fit of the data pair ($\ln y_i$, x_i) yields an intercept of ln A and a slope of m.

The functional form for the hot-wire velocity probe can also be determined with the linear least squares technique using the data pair (V_i^2, $\sqrt{v_i}$). Thus much can be achieved using this simple least squares technique.

9.3 Exercise 3. Calibration and Conversion of Transducer Data

The purpose of this exercise is to calibrate a transducer, take data for an experiment and then produce plots of the transducer output converted from voltage to the quantity measured. For this exercise you will use the data processing capabilities implemented as choices in the "utility" package. While you don't need to exit the program to exercise these routines the analysis is viewed as "out-of-line" in that the calculations are not done while the data is taken. Section 14 shows how to perform online linear conversion using the conversion constants.

In this section you will also review how to retrieve files as input to the UnkelScope software package and learn the format of the stored data.

The example chosen is the calibration of a pressure transducer, using a standard dial type gage as the standard. This is only illustrative and in no way represents a limitation of the software to pressure transducer measurements.

Additional Equipment. In addition to the computer-based system and your experiment (with a transducer), you will need a standard against which this transducer can be calibrated and a way to generate steady levels of the physical variable over the appropriate range of the transducer.

9.4 Overall Process

The overall procedure will be:

a) Set up the calibration standard and the transducer.

b) Using the UnkelScope program, take the transducer voltage data for a static calibration process.

c) Store the data in a disk file.

d) Run the calibration routine to get the curvefit parameters.

e) Display the fitting curve and the calibration data.

f) Take a set of data from the experiment.

g) Use the conversion program from the Utility command line to convert from voltage to the quantity of interest.

h) Display the reduced data with the UnkelScope program.

While this may appear tedious, the effort required is much less than would be necessary to do these things by hand. Once the calibration constants are known, online conversion can also be done (see Section 14).

9.5 Taking the Calibration Data

9.5.1 Setup and Planning. Setup and check out your transducer and source of signal for the transducer. You should plan a sequence of values that:

a) covers as much of the operating range as possible;

b) allows for repeated points; and

c) avoids a strictly monotonic variation in the points.

You will be taking 16 samples of the transducer output at each condition. The calibration program supports a set of up to 20 conditions which is more than adequate in most cases.

You should become familiar with the equipment that provides the input to the transducer and should know how long it takes for the system to reach steady conditions. Figure 9.2 shows the setup to perform a calibration of a pressure transducer. A pressure source is used to adjust the pressure applied to a 'standard' gage and the transducer.

9.5.2 Sampling the Data. The calibration program requires a particular format of the data. It expects the transducer output to be stored in Vertical Trace 1 and that there will be 16 values of the transducer output at each level of the transducer input. To do this conveniently you should use the keyboard to trigger each sample. This triggering is accomplished by selecting the Trigger Mode of Single/Trg/Sync and a Trigger Source of Keyboard. In this mode you get one sample rather than one scan for each trigger signal. To provide the most information while doing the calibration, set the time between samples to 1 second so that the digital value is written to the display screen. Note that the program does NOT take a sample when the trigger is received but waits for a time equal to the time between samples. So if you press the keyboard 16 times quickly you will still have to wait 16 seconds for all samples to be taken.

For each "point" in the calibration you should:

a) set the transducer input and wait for steady state;

b) initiate 16 samples by pressing the space bar 16 times; and

c) record the value measured by the standard for later use.

When you have taken all the desired points, press ESC to abort the scan. Then, respond to questions in the message line to return to the command mode. Store the data file and make note of the file name.

Figure 9.3 shows the data and setup configuration used for this example. 9 points were taken with 2 repeats of points.

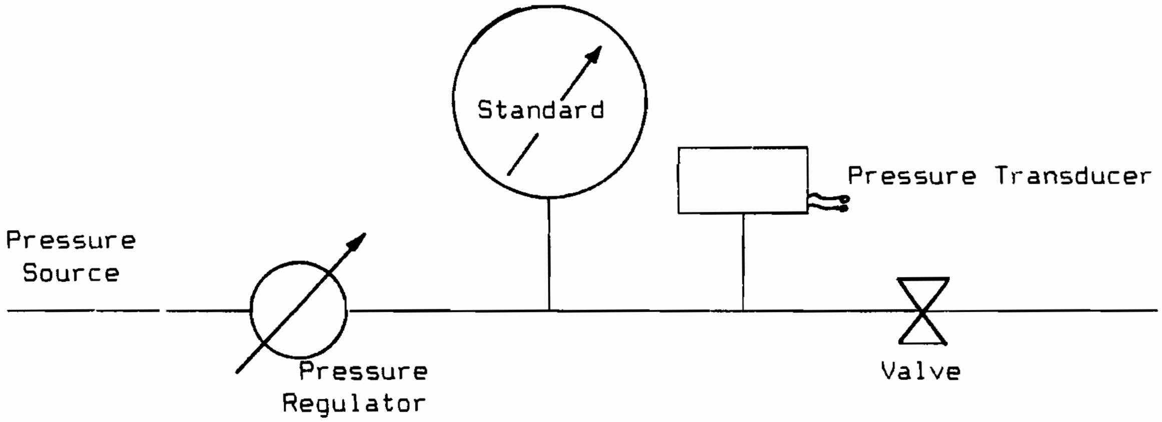


Figure 9.2 Schematic of setup to calibrate a pressure transducer.

```
------- Vertical Trace 1 -------        --- Additional Vertical Traces ---
Source [Analog  0] A/D Range [+/- 10]   Tr Input                      A/D
Label  :Pressure Signal                 #  Chan  Label:             Range
Span   [  5   v full scale]             3 [none ]
Range  [  .00E+00 to  5.00E+00]         4 [none ]

------- Vertical Trace 2 -------
Source [None      ]

                                        ----------- Triggering -----------
                                        Mode  [Sngl/Pt Syn] Source [Keyboard ]

------- Horizontal Trace -------
Source [Time      ]                     ----------- Processing -----------
Label  :Time(Seconds)                   Type [none                    ]
Span   [200  s full scale]
Range  [  .00E+00 to  2.00E+02]

----------- Sampling -----------
Sample Rate [  1 s          1 hz]
[ 256] Samples  (Scan Time 2.56E+02 s)
(Real Time Plot) (Processing Active  )
```

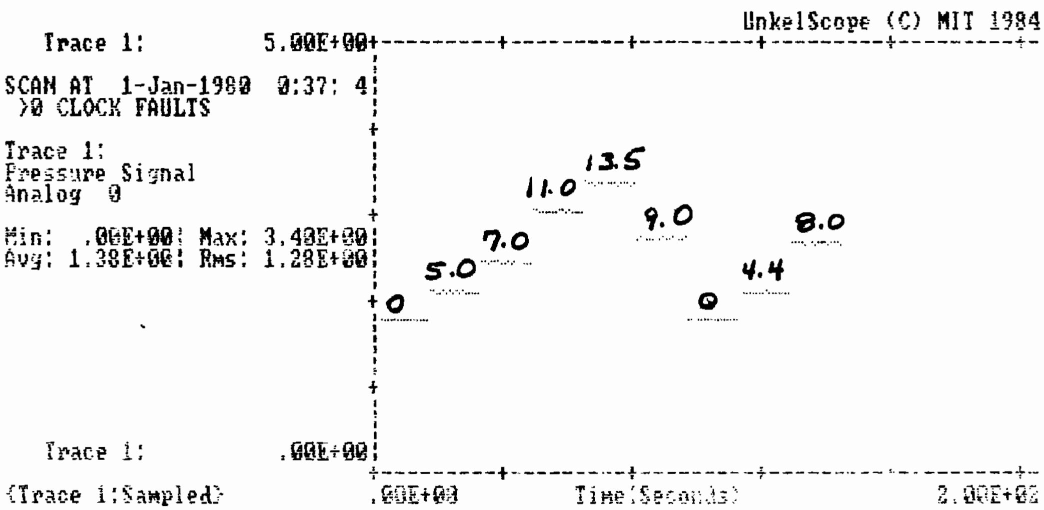


Figure 9.3. Setup and data typical of calibration process.

9.6 Generating the Curvefit Parameters and Examining the Results

9.6.1 Using the Calibration Utility. The next step is to run the calibration utility routine which determines the curvefit constants. This program also stores a data file on the disk to allow you to display the calibration data and fitting curve on the same graph. The Calibration routine requires 5 inputs:

a) the file name storing the calibration data file;
b) the file name where reduced data will be stored;
c) the label describing the physical quantity (and units);
d) the number of calibration points; and
e) the set of values of transducer inputs in the proper order.

The routine prompts for each of these inputs. To run the routine move to the UTILITIES choice in the command mode. Then move to the Calibration choice in the Utility Command Mode.

Respond to the questions as described below and record the final results as necessary. The program prints the curvefit equation in both directions - i.e. from the transducer output to the parameter measured and vice versa. On the IBM PCXT you can get a hardcopy of the full dialogue by making an image of the screen when you are done (press shift/prtscr). Figure 9.3 shows typical image of the screen. When you finish you will be back in the command mode pointing to the UTILITIES choice.

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

Inputs to The Calibration Utility

Input File Name: Enter the name of the file that contains the data to be used as the input. This data must be in the format described in Section 9.5. Use delete and Ctrl/U to fix mistakes. If the requested file does not exist, an error message will be printed in the upper left corner. You will be given an opportunity to examine the contents of the current directory and then an opportunity to try a different file name. When the file is read, the label from the file is printed and some simple statistics about the file may be printed in the lower portion of the screen.

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

Label for the Physical Variable: This is the label that defines the quantity being measured i.e., pressure. This label will be used as the vertical label. The horizontal label will be taken from the data file.

Settings in Calibration(2 to 20): Input the total number of conditions in your set of calibration data. At least 2 points are required to fit a straight line.

Curvefit Type If your version allows curvefits higher than linear, the program will let you toggle to the appropriate choice. If your version handles only linear curvefits, this line will display [Linear] and not stop at this position.

Settings. For each of the conditions, the program will print the average value of the transducer output and then will highlight a 9 character field where you can enter the physical variable at that setting in the calibration process. When all the settings have been input, a message will be printed at the bottom of the screen asking you whether you wish to accept the values shown. If you do not wish to accept them, you can move "up" to any one you wish to correct. If you do wish to accept them press return and the curvefit will be performed.

Figure 9.4 shows the output from the calibration routine.

```
           Input file name:press.dat
      Label from Input File:Pressure Calibration 1-Jan-1980  0:37: 4
          Output file name:press.cal
      Label for Output File:Pressure Calibration

     Least Squares CurveFit of Calibration Points In Trace 1
         Label for Physical Variable:Pressure (psig)
     Settings in calibration(2 to 20): 9.00E+00
                      Curvefit Type [Linear   ]
     point transducer physical
            reading    value
      1    1.79E+00     .00E+00
      2    2.11E+00    5.00E+00
      3    2.45E+00    7.00E+00
      4    3.04E+00    1.10E+01
      5    3.37E+00    1.35E+01
      6    2.74E+00    9.00E+00
      7    1.77E+00     .00E+00
      8    2.10E+00    4.40E+00
      9    2.68E+00    8.00E+00
       physical value=  8.156E+00 * transducer output  -1.355E+01
     transducer output=  1.226E-01 *    physical value + 1.662E+00
```

Figure 9.4. Output from the calibration utility.

Exiting and Obtaining Hardcopy. When the last point is entered the program stores the file and prints the message: "Press Y to Return to Command Mode." On the IBM PCXT you can get a hardcopy of the screen by typing Shift/PrtSc, then type Y to return to the command line pointing to the UTILITIES choice.

9.6.2 Displaying and Evaluating the Results.

One effect of the calibration program is to store information in the calibration file so you can display the calibration data as a plot of physical value vs. transducer reading. With this you can compare the calibration curve directly to the data. To accomplish this the calibration utility stores the "line" in Vertical Trace 2 in the file and stores the "data" in Vertical Trace 1.

Note that the physical parameter is plotted on the vertical scale and the transducer output value on the horizontal scale.

Retrieving the Data. To retrieve the data in the processed calibration file, toggle to the [Fil Ch 1] (read as File Channel 1) choice for the Vertical Trace 1 Source. This means to select, as the source of the current Vertical Trace 1, the contents of Vertical Trace 1 as stored in the file. Enter the name of the file that was specified as the output file of the calibration program and press the down arrow key. The program will read the data file, print the label from the file in the message line and put the Vertical Trace 1 label stored in the file Vertical Trace Label option line. Choose an appropriate range for the expectd values of the physical parameter (pressure in this case.)

After retrieving the input for Vertical Trace 1 adjust the span and range as needed. Then move to the Vertical Trace 2 Source option line. Toggle to the [Fil Ch 2] choice (to choose Trace 2 as stored in the file as the input to Vertical Trace 2), and enter the file name. Then adjust the span and range to be the same as those for Vertical Trace 1.Trace 2 of the file written by the calibration utility represents the curvefit to the data.

Selecting the Horizontal Trace. Move to the Horizontal source option and toggle to select the [Anlg 1] option. Then choose a span and range appropriate to the transducer output voltage range which will be plotted as the horizontal axis.

Note that the label shown for the horizontal trace applies only to newly sampled data; the proper label from the file will appear on the plot. The time between samples, triggering option and special processing are irrelevant when displaying previously stored data only.

Display Results. Move to the SMPL/DSPLY option and press the down arrow key. The result should be a plot of the data and curvefit line. Note that the plot is in the form of physical value vs. transducer output voltage. Return to the setup mode and adjust the scales as necessary, then make a hardcopy of the results by executing PRNT DSPLY.

Figure 9.5a and 9.5b show the plotted results for the calibration data of Figure 9.3 and 9.4.

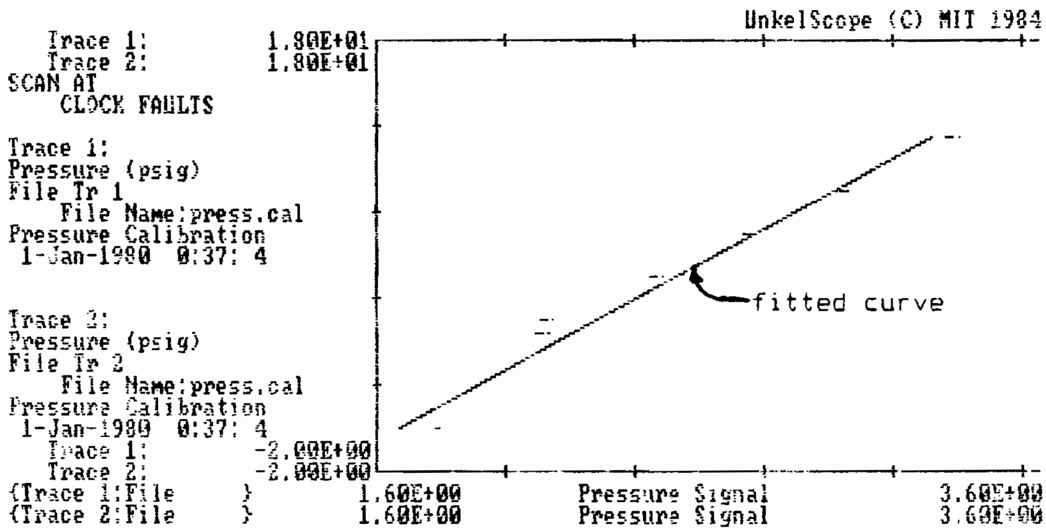


Figure 9.5a. Plotted results of the calibration process. Note the large discrepancy between the 0 psig point and the remaining points. To obtain a good linear fit this problem must be resolved (see Figure 9.5b.)

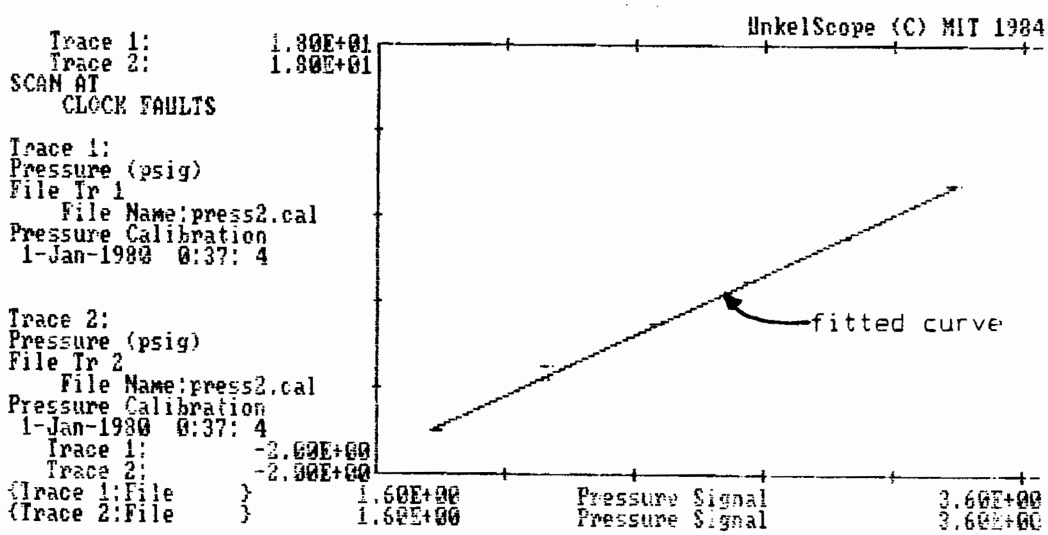


Figure 9.5b. Redone calibration results with problem corrected.

9.7 Converting A Set of Data

Take a set of data according to the needs of your experiment and store the data in a file. The conversion utility will allow you to convert each element of the data file. For each trace stored by the software, 3 arrays of data are retained. The first array contains the vertical trace values and the third array the corresponding times. The second array may contain the "horizontal" trace variables, which in many cases is simply a repeat of the time. In other cases however, the horizontal trace contains the values against which the vertical trace has been "cross-plotted," i.e., it may contain the sampled signal. Run the conversion utility by moving to the UTILITIES choice in the command line and then the Conversion choice in the Utilities Command Mode.

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

Inputs to The Conversion Utility

Input File Name: Enter the name of the file that contains the data to be used as the input. Use delete and Ctrl/U to fix mistakes. If the requested file does not exist, an error message will be printed in the upper left corner. You will be given an opportunity to examine the contents of the current directory and then an opportunity to try a different file name. When the file is read, the label from the file is printed and some simple statistics about the file may be printed in the lower portion of the screen.

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

For each valid channel the program will prompt for changes in each of the 3 arrays of data. The trace number and trace label will be printed. Then you will be allowed to decide whether to convert the trace or not. If you do not wish to convert that trace, press the down arrow key to move to the next signal. If you do wish to convert the trace, toggle to [yes] and press the down arrow key.

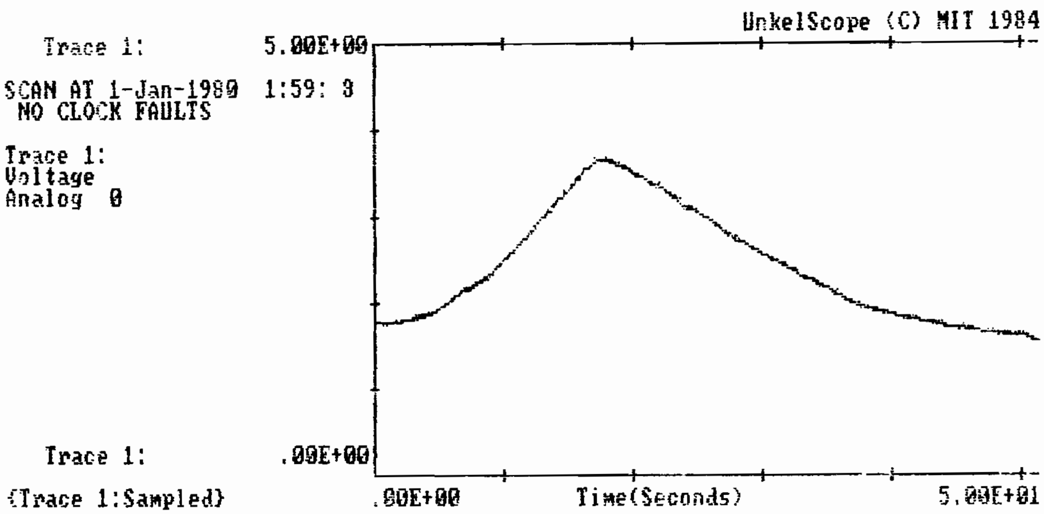
For each signal you convert you will make the following additional inputs:

<u>Modify Label.</u> The old label can be modified by deleting characters and entering new ones.

<u>Conversion Equation Constants.</u> The conversion is represented by the slope and intercept. Each are specified in the 9 character field as described above, with the slope entered first and then the intercept.

Exiting and Obtaining Hardcopy. When your choices for the last signal are entered, the program stores the file and prints the message "Press Y to Return to Command Mode." On the IBM PC you can get a hardcopy of the screen by typing Shift/PrtSc. Type Y to return the command line of the UnkelScope program with the cursor pointing to the UTILITIES choice.

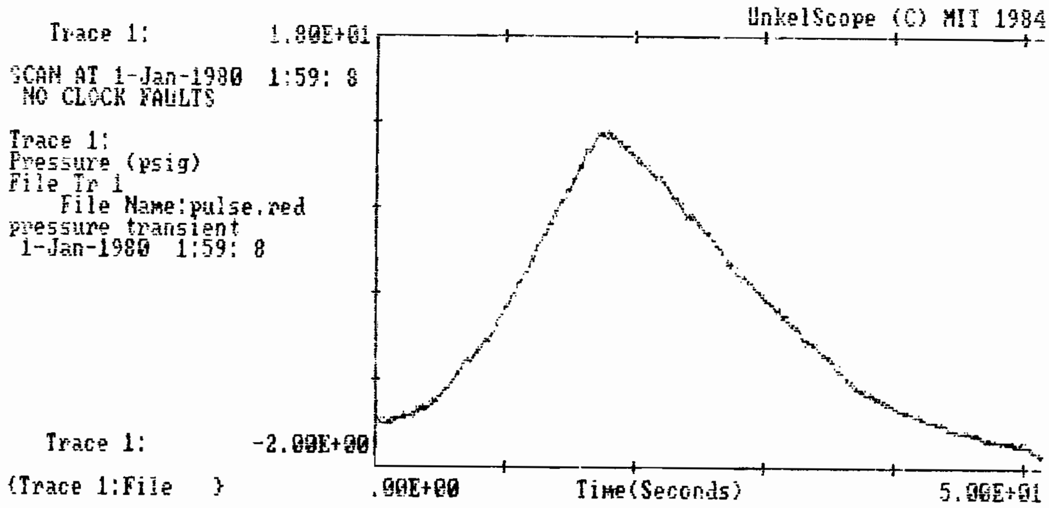
Figure 9.6 shows a plot of the raw data and a sample of the dialogue for using the conversion utility.




```
Conversion
      Input file name:pulse.dat
Label from Input File:pressure transient   1-Jan-1980  1:59: 8
     Output file name:pulse.red
Label for Output File:pressure transient

Trace  Change?  Label                  =Old Value *    slope    +    intercept
Vert 1  [yes] Pressure (psig)          =Old Value * m: 7.07E+00 + b: -1.24E+01
Horz 1  [ no] Voltage
Time 1  [ no] Time(Seconds)
```

Figure 9.6 Hardcopy of sample dialogue for the conversion utility and plot of raw data.

<u>Making The Plot.</u> The plot of the reduced data is obtained by modifying the Setup configuration to read the appropriate data file and set the appropriate scale settings. Figures 9.7a and 9.7b show the setup and plotted results for the original reduced traces of the transducer output.




```
                                               UnkelScope (c) MIT 1984
Commands        QUIT       SETUP MODE   SMPL/DSPLY   SAVE SETUP   GET SETUP
             SAVE DATA     PRNT SETUP   PRNT DSPLY   UTILITIES

--------- Vertical Trace 1 ---------     ------ Additional Vertical Traces ------
Source [File Tr 1]File Name:pulse.red    Tr Input                          A/D
Label  :Pressure (psig)                  #  Chan  Label:                   Range
Span   [ 20    v full scale]             3 [none ]
Range  [-2.00E+00 to  1.80E+01]          4 [none ]

--------- Vertical Trace 2 ---------
Source [None      ]

                                         ------------ Triggering ------------
                                         Mode  [Singl Sweep] Source [Keyboard ]
--------- Horizontal Trace ---------
Source [Time      ]
Label  :Time(Seconds)                    ------------ Processing ------------
Span   [50    s full scale]              Type [none                        ]
Range  [  .00E+00 to  5.00E+01]

------------- Sampling -------------
Sample Rate [ 50 ms        20 hz]
[1024] Samples  (Scan Time 5.12E+01 s)
(Real Time Plot) (Processing Active  )
```

Figure 9.7a,b. Setup configuration and plot of data converted from a pressure transducer.

9.8 Summary

In this exercise you have learned how to use the program to calibrate a transducer and to convert data from the originally stored voltage levels to the values of the parameter measured. In the programs implemented here a linear calibration curve was assumed and the data compared graphically against this curve to evaluate the "goodness" of the fit.

As part of the exercise you have reviewed how to access previously stored files of data.

10.0 Exercise 4: Point Plotting

This utility allows data input by the keyboard to be plotted as a sequence of symbols at specified locations. The goal is not to provide final copy plots but quick results that can be traced to get final draft material if necessary. The program allows entry of up to 20 points and stores these points in a file that can later be plotted using the Scope program. Labels are also stored and will be printed with the plotted copy. Only 1 "set" of data can be stored for each run of the Point Plot Utility.

10.1 Running the Program

Move (using the left/right arrow keys) to the UTILITIES choice of the command line and press the down arrow key to enter the Utility Command Mode. Toggle to the Point Plot choice and execute it by pressing the down arrow key.

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

Inputs to The Point Plot Utility

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

Symbol is a [. Toggle to the choice of symbol type from those available.

Vertical Trace Label: This is the label that applies to the variable plotted on the vertical axis.

Symbol Size in Vertical Direction: The program sets the symbol size in terms of the variable rather than as a fraction of the axis spacing when it is plotted. If you wish the "points" to be of reasonable size on the final plot choose them to be 1/40th of the span between the minimum and maximum points. Otherwise pick the size to be the "error" size.

Horizontal Trace Label: This is the label that applies to the variable plotted on the horizontal axis.

Symbol Size in Horizontal Direction: Same as above but for the x variable.

Number of Points (maximum is 20): The total number of "points" to be plotted; if you set it greater than 20 the program will reset the value to 20.

Pairs of (vertical, horizontal) Enter the point locations vertical value first then horizontal value. Use up to 9 characters to define a number. After it is entered the number is converted to FORTRAN "e" format and rewritten. Program does its best when incorrect characters are typed.

After the numbers have been input, the program prints a message at the bottom of the screen asking you whether you wish to accept the values as they are. If you do wish to accept the values, press the down arrow key. If you do NOT wish to accept the values, use the up arrow key to move "up" to the value(s) you wish to modify. Only the point values can be changed.

Exiting and Getting Hardcopy. When the last point is entered the program stores the file and prints the message: "Press Y to Return to Command Mode." On the IBM PCXT you can get a hardcopy of the screen by typing Shift/PrtSc. Type Y to the command line and points to the UTILITIES choice.

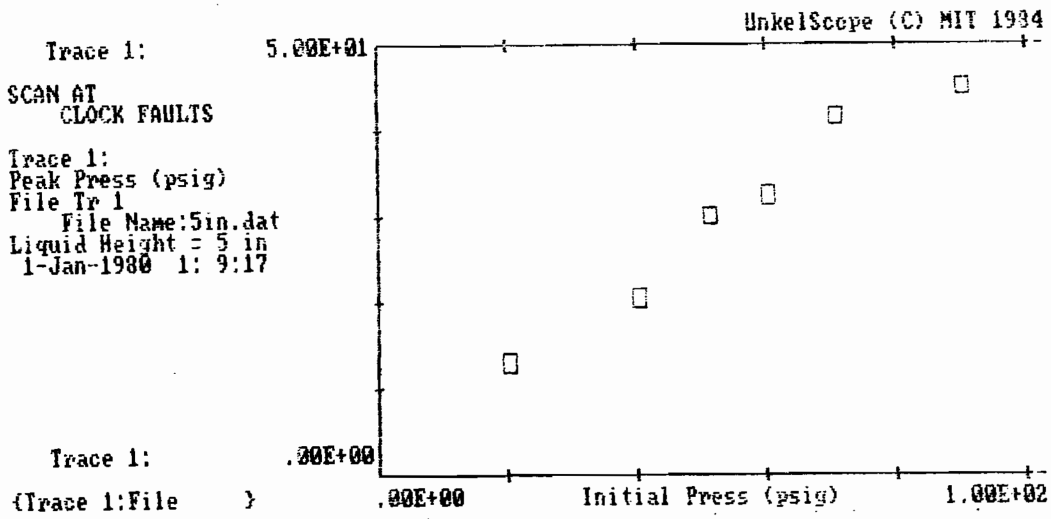
Figure 10.1 shows a hardcopy of the dialogue for the Point Plot Utility.

```
                                                   UnkelScope (c) MIT 1984
Point Plot


     Output file name:5in.dat
Label for Output File:Liquid Height = 5 in
                          Symbol is a [Rectangle]
                   Vertical Trace Label:Peak Press (psig)
   Symbol Size in Vertical Direction: 2.00E+00
                 Horizontal Trace Label:Initial Press (psig)
  Symbol Size in Horizontal Direction: 2.00E+00
      Number of Points(maximum is 20): 6.00E+00

Pairs of(vertical,horizontal)9 characters each number with return in between
---- 1---> 1.30E+01   2.00E+01
---- 2---> 2.05E+01   4.00E+01
---- 3---> 3.24E+01   6.00E+01
---- 4---> 4.50E+01   9.00E+01
---- 5---> 3.00E+01   5.10E+01
---- 6---> 4.15E+01   7.05E+01



        Press Return to Accept Values. Use Up Arrow to Edit Values.
```

Figure 10.1 Hardcopy of the dialogue made while running the point plot utility.

10.2 Making the Plot

The plot is made by a) entering SETUP MODE to chose the setup parameters appropriately and b) executing the option PRNT DSPLY. First select Trace 1 of the file you stored. Toggle the source of Trace 1 to read [Fil Tr 1], press return and then enter the file name. Then set the proper span and range. Select the name for Vertical Trace 2 source. Select "[Analg 0]" for the Horizontal Trace source and then set the horizontal span and range. Ignore the label listed for Horizontal Trace; the proper label will come out on the plot. Figure 10.2.a shows the setup parameters for the plot shown in Figure 10.2.b




```
Vertical Trace 1
Source [File Tr 1]File Name:5in.dat
Label  :Peak Press (psig)
Span   [ 50   v full scale]
Range  [   .00E+00 to  5.00E+01]

Additional Vertical Traces
Tr Input                          A/D
#  Chan   Label:                  Range
3 [none ]
4 [none ]

Vertical Trace 2
Source [None      ]

Triggering
Mode  [Sngl/Pt Syn] Source [Keyboard ]

Horizontal Trace
Source [Analog  0]  A/D Range [+/- 10]
Label  :
Span   [100   v full scale]
Range  [  .00E+00 to  1.00E+02]

Processing
Type [none                        ]

Sampling
Sample Rate [  1 s          1 hz]
[ 256] Samples  {Scan Time 2.56E+02 s}
{Real Time Plot} {Processing Active  }
```

Figures 10.2a and b. Setup and display typical of a point plot.

11.0 Statistical Manipulation of Data

The quality of a signal can often be improved by using statistical processing devices such as spectral sensitive filters. The most common example is "smoothing" data by passing it through a digital low pass filter. High pass and band pass filters are also useful. The Fast Fourier Transform can also be used to advantage by displaying the spectral character of a signal or by determining the correlation between signals. The UnkelScope program supports some of these techniques as out-of-line data processing utilities. This section describes these statistical functions and demonstrates their application. The discussion is not meant to be rigorous, but rather illustrative, showing the usage and identifying the conditions under which difficulties arise. The next section discusses the filters; the final section discusses FFT techniques.

11.1 Digital Filters

A filter is used to modify a signal by selectively altering the magnitude of a signal at particular frequencies. Filters can be analog or digital, i.e., can operate before or after the signal has been digitized; analog filter design is not covered here but can also be implemented. The character of a filter is described by the attenuation (gain) and phase shift of the filter as a function of input signal frequency. For example, a "perfect" low pass filter designed to pass frequencies lower than 10 Hz would have the frequency dependent performance shown in Figure 11.1. The filter would reproduce exactly all components of the input with frequency less than 10 Hz and would eliminate completely those components above 10 Hz. A real filter has a less sharp cutoff and introduces phase shifts between the input and output. A reasonably designed low pass filter characteristic is shown in Figure 11.2. The terminology of the filter is described in this figure. The "cutoff" frequency is 10 Hz. At low frequency there is no alteration of the signal, while at high frequency the signal is strongly attenuated. There is a slight peak just before the cutoff frequency and there is some peculiar behavior at high frequency. The high frequency "side-lobes" can result in difficulties in signal interpretation but usually are not important. The filter used in UnkelScope is a 2nd order Butterworth filter and is discussed in detail in several texts.

To compute the filtered values of the first few sampled points, some assumptions must be made about what the signal was doing prior to the time the first data point was taken. In this software, the signal is assumed to have been at a constant value equal to the value of the first sampled point. This avoids sharp changes in level, but in most instances results in a discontinuous value of the slope.

The next subsections show how to use the software interface and then show by example how the filters improve the quality of some representative signals.

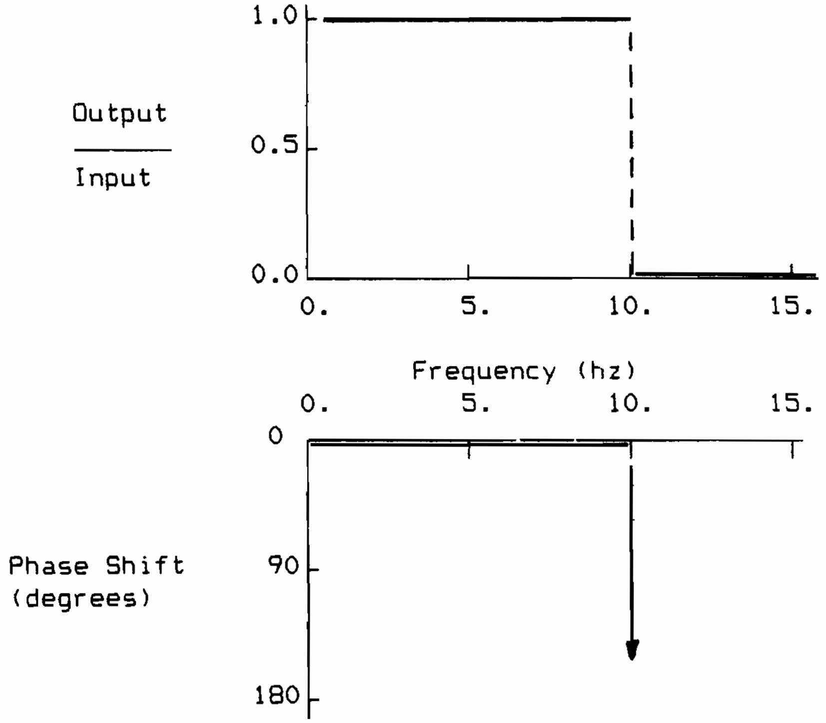


Figure 11.1 Gain and phase shift versus frequency for a "perfect" low pass filter.

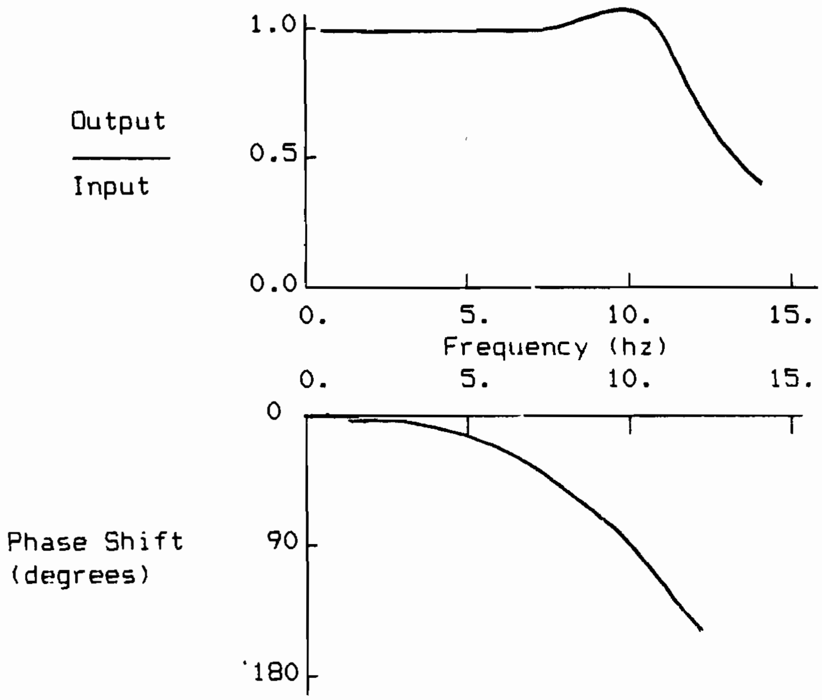


Figure 11.2 Gain and phase shift versus frequency for a digital low pass filter.

11.1.1 Applying the Filters

Move (using the left/right arrow keys) to the UTILITIES choice of the command mode and press the down arrow key to enter the Utility Command Mode. Toggle (using the left/right arrow keys) to the STATISTICAL choice and execute it by pressing the down arrow key. Note that the questions are sequential and there is no "editing" by moving back through the questions (i.e., the input is not handled in the same manner as the Setup Mode of the program.)

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

The Inputs for Statistical Processing Using Filters

Input File Name: Enter the name of the file that contains the data to be used as the input. Use delete and Ctrl/U to fix mistakes. If the requested file does not exist, an error message will be printed in the upper left corner. You will be given an opportunity to examine the contents of the current directory and then an opportunity to try a different file name.

When the file is read, the label from the file is printed and some simple statistics about the file may be printed in the lower portion of the screen.

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

Input File Name: Enter the name of the file containing the data you wish to filter Use Delete and Ctrl/U to fix mistakes.

Selection is allowed sequentially for each valid channel. The program assumes the measurements to be spaced equally in time.

<u>Modify Label.</u> The old label for the signal will be printed. You can modify this label if you wish.

<u>Operation.</u> Use the left and right arrow keys to toggle to the desired choice and the down arrow key to move to the next selection. The filter choices are high pass, low pass and band pass.

<u>Cutoffs.</u> For the low pass filter the low cutoff must be entered. For a high pass the high frequency cutoff must be specified. For the band pass the low pass and high pass frequences must be entered (remember the frequencies passed will be between the low and high cutoffs specified). The numbers are entered as described earlier.

<u>Exiting and Obtaining Hardcopy.</u> When the last signal has been processed, the program stores the file and prints the message: "Press Y to Return to Command Mode." On the IBM PC you can get a hardcopy of the screen by typing Shift/PrtSc. Type Y to return to the command line.

11.1.2 Low Pass Filter to Reduce "Noise"

Consider the signals shown in Figure 11.3 that might represent data taken using a sensor in a very "noisy" environment. It is extremely important to understand where the noise comes from and to make sure that a digital filter is the appropriate solution to the problem. There are at least 3 ways to reduce an unwanted noise component:

a) reduce or eliminate the source of noise;
b) use an analog filter to reduce the noise; and,
c) use a digital filter to reduce the noise component.

Given equal difficulties in implementing all three solutions, reducing the noise at the source is the best solution. If the frequency of the noise is 60 Hz or a multiple of 60 Hz the noise source is likely to be a "ground-loop" or other electrical problem. Ground loops should be eliminated and in many cases are symptoms of more serious instrumentation problems.

If the noise source cannot be eliminated, the other solutions can be applied. In either case it is important that the desired component of the signal is NOT affected by the filtering process. At the very least the filtered and unfiltered signals must be compared for a reasonable number of cases to show that the signal is not adversely affected by the filter. The digital filter has the advantage that it is possible to reprocess a recorded signal until the proper choice is made. A further advantage of the digital filtering over analog filtering is obtained if the relationship between the physical parameter measured and the transducer output voltage is non-linear. In such a case the individual voltage readings can be converted to the physical variable first and the filter applied to the time varying signal for the physical variable. With the above comments in mind, the result of applying a low pass filter at each of 3 frequencies is now presented.

For convenience the same signal has been measured as input to all 4 analog input traces so that the signal can be processed in different ways. For this example the high frequency component is not of interest i.e., it is noise. Figure 11.4a shows a hard copy of the selections for this exercise. Trace 1 was left unaltered while filter cutoffs of 200 Hz, 20 Hz and 2 Hz were applied to trace 2, 3 and 4 respectively. The displayed results are shown in Figures 11.4b and 11.4c. It is clear that the 200 Hz filter does little to improve the signal quality while the 2 Hz filter has degraded the main part of the signal. It is important to make such comparisons of input and output when applying the filters.

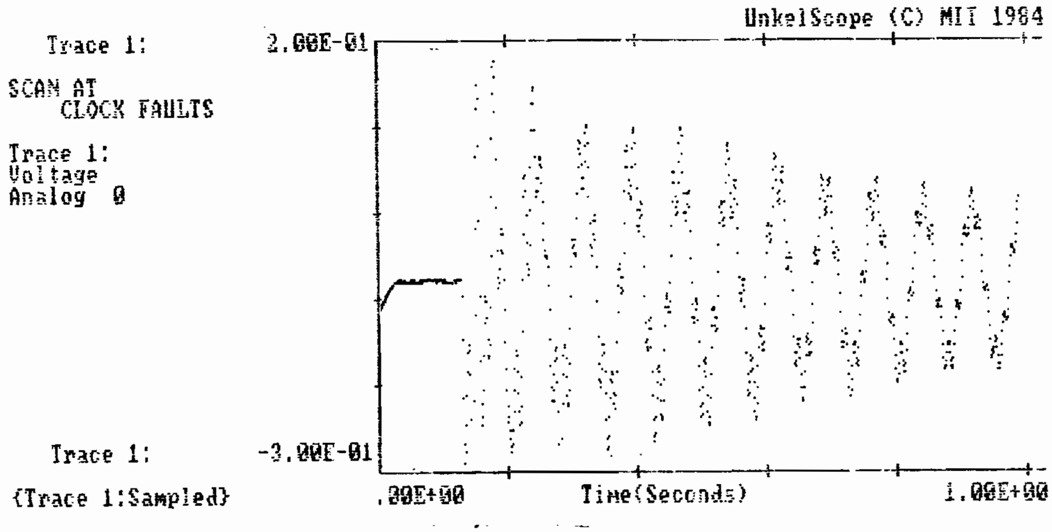


Figure 11.3 Measured sensor data in a noisy environment.

UnkelScope (c) MIT 1984

Statistical

Input file name:noisy.raw
Label from Input File:noisy sensor data
Output file name:noisy.dat
Label for Output File:filtered noisy data

Trace Label	Operation	Parameters
Vert 1:Voltage	[none]	
Horz 1:Voltage	[none]	
Vert 2:Voltage	[Low Pass]	Pass Below : 2.00E+02
Horz 2:Voltage	[none]	
Vert 3:Voltage	[Low Pass]	Pass Below : 2.00E+01
Horz 3:Voltage	[none]	
Vert 4:Voltage	[Low Pass]	Pass Below : 2.00E+00
Horz 4:Voltage	[none]	

Figure 11.4a Screen display of selections for Exercise 5a showing the use of the low pass filter.

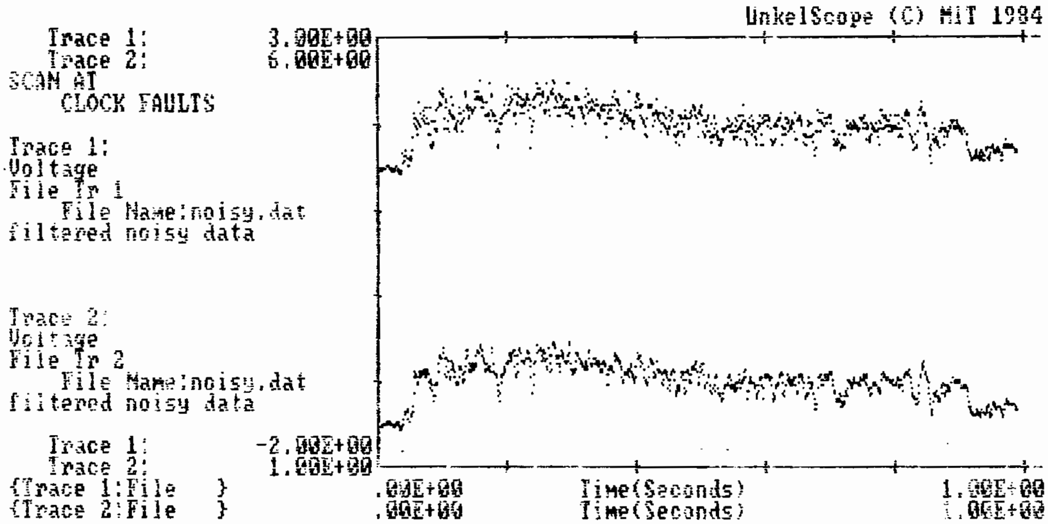


Figure 11.4b Results showing the unfiltered signal and the signal after it has been passed through a 200 Hz low pass Filter.

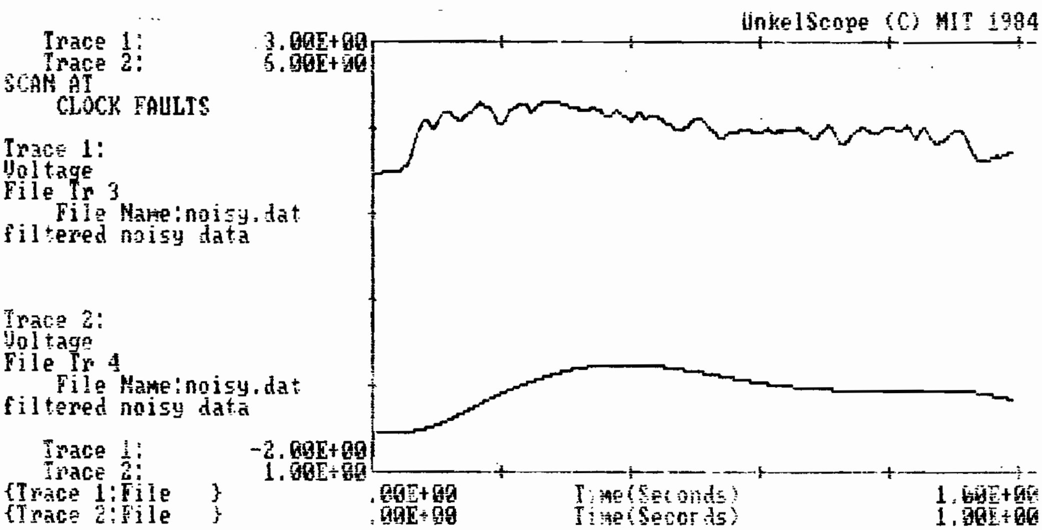


Figure 11.4c Displayed result showing the signal filtered by a 20 Hz and 2 Hz digital filter.

11.1.3 High Pass Filter to Remove Drift

In some circumstances a transducer will have a low frequency drift superimposed on a high frequency signal of interest. The high pass filter can be used to improve the signal quality in such a case. For convenience the same signal has been measured as an input to all 4 traces. These are then processed differently and redisplayed. Figure 11.5a shows a hardcopy of the selections made in this case. Trace 1 was left unaltered while filter cutoffs of 2 Hz, 20 Hz and 200 Hz were applied to the signal in Traces 2, 3 and 4 respectively. The displayed results are shown in Figure 11.5b and 11.5c. As can be seen the 2 Hz filter cutoff is too low - i.e., there is still a noticeable drift over a good portion of the signal. The higher value for cutoff (200 Hz) is too high and the entire signal itself is affected.

```
Inputs Completed. Working                               UnkelScope (c) MIT 1984
Statistical
      Input file name:spread.raw
Label from Input File:drift on oscillation
     Output file name:spread.dat
Label for Output File:example of high pass

Trace  Label                  Operation       Parameters
Vert 1:Voltage                [none          ]
Horz 1:Voltage                [none          ]

Vert 2:Voltage                [High Pass     ] Pass Above : 2.00E+00
Horz 2:Voltage                [none          ]

Vert 3:Voltage                [High Pass     ] Pass Above : 2.00E+01
Horz 3:Voltage                [none          ]

Vert 4:Voltage                [High Pass     ] Pass Above : 2.00E+02
Horz 4:Voltage                [none          ]
```

Figure 11.5a Screen display of selections for Exercise 5b showing the use of the high pass filter.

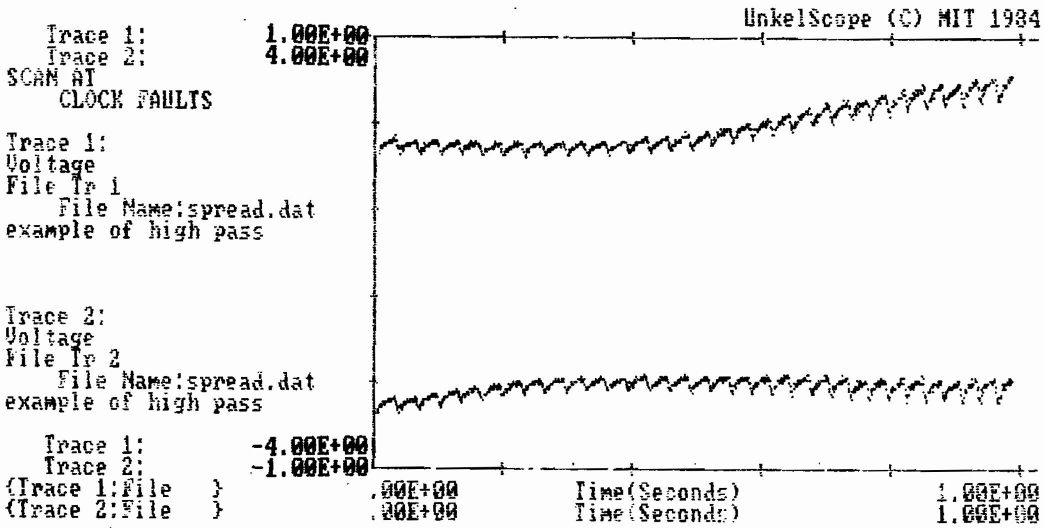


Figure 11.5b Result showing the unfiltered signal and the signal after it has been passed through a 2 Hz high pass filter.

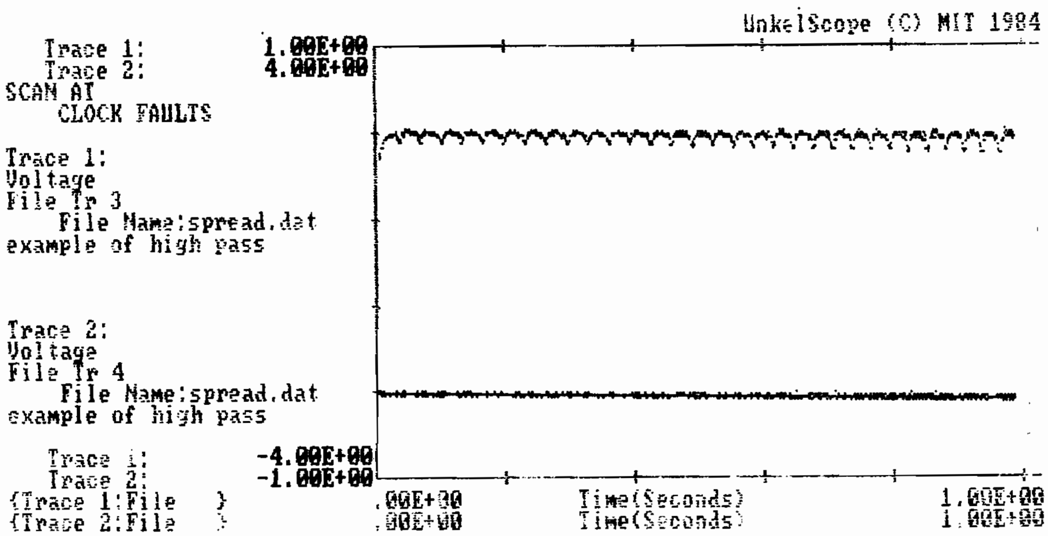


Figure 11.5c Result showing the signal after it has been passed through a 20 Hz and 200 Hz high pass filter.

11.1.4 Band Pass to Select Known Frequency Component

Another situation where signal processing can be useful is when a system is excited at a known frequency and only the response at that frequency is desired. If the excitation frequency is fixed over the whole sample time, a band pass filter can be used to improve the "signal-to-noise" ratio of the measurement. A lock-in amplifier is the analog counterpart of this technique. Consider the signal shown in Figure 11.6 which looks very "noisy." Consider now that the frequency of the desired effect is at 140 Hz. For convenience, the same signal has been measured as input to all traces. Figure 11.6a shows a hard copy of the selections made for this example. Trace 2 was processed by a band pass between 50 Hz and 225 Hz. Trace 3 was processed by a band pass between 105 Hz and 185 Hz. Trace 4 was processed by a band pass filter between 200 Hz and 280 Hz. The results are displayed in Figure 11.6b and 11.6c. Note the great improvement in both Trace 2 and 3 (Figure 11.6c) as the noise is reduced and the signal emerges. In the last case (lower trace of Figure 11.6c) a band pass with the wrong center frequency was chosen and the signal is lost. Also note the longer 'relaxation' time as the band pass is narrowed.

```
                                              UnkelScope (c) MIT 1984
 Statistical
       Input file name:jetpul.raw
 Label from Input File:interrupted jet flow
      Output file name:jetpul.dat
 Label for Output File:example of band pass

Trace  Label                      Operation        Parameters
Vert 1:Voltage                    [none         ]
Horz 1:Voltage                    [none         ]

Vert 2:Voltage                    [Band Pass    ]  Pass From : 5.00E+01 to: 2.25E+02
Horz 2:Voltage                    [none         ]

Vert 3:Voltage                    [Band Pass    ]  Pass From : 1.05E+02 to: 1.85E+02
Horz 3:Voltage                    [none         ]

Vert 4:Voltage                    [Band Pass    ]  Pass From : 2.00E+02 to: 2.80E+02
Horz 4:Voltage                    [none         ]
```

Figure 11.6a Screen display of selections for Exercise 5c showing the use of the band pass filter.

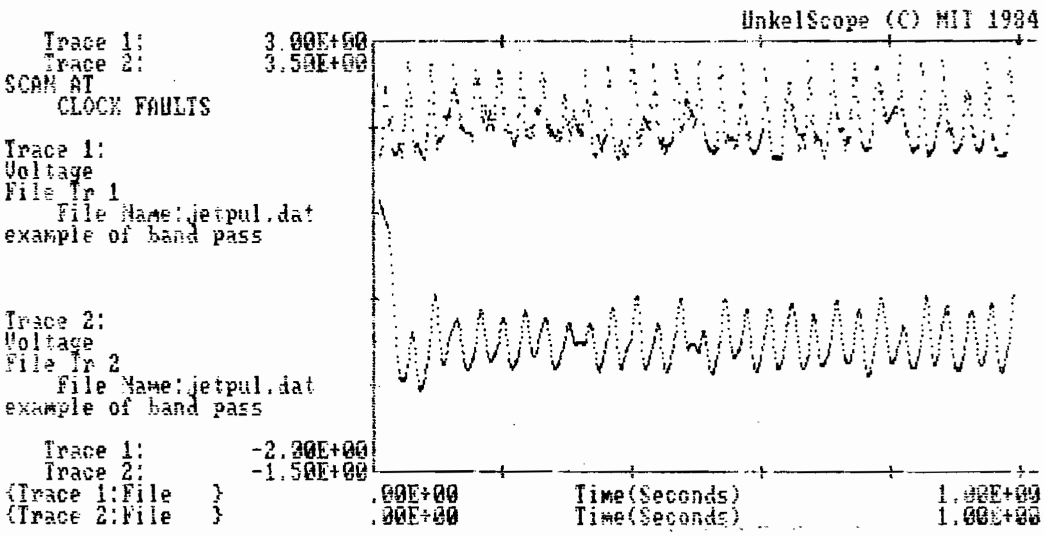


Figure 11.6b Result showing the unfiltered signal and the signal after it has been passed through a band pass filter.

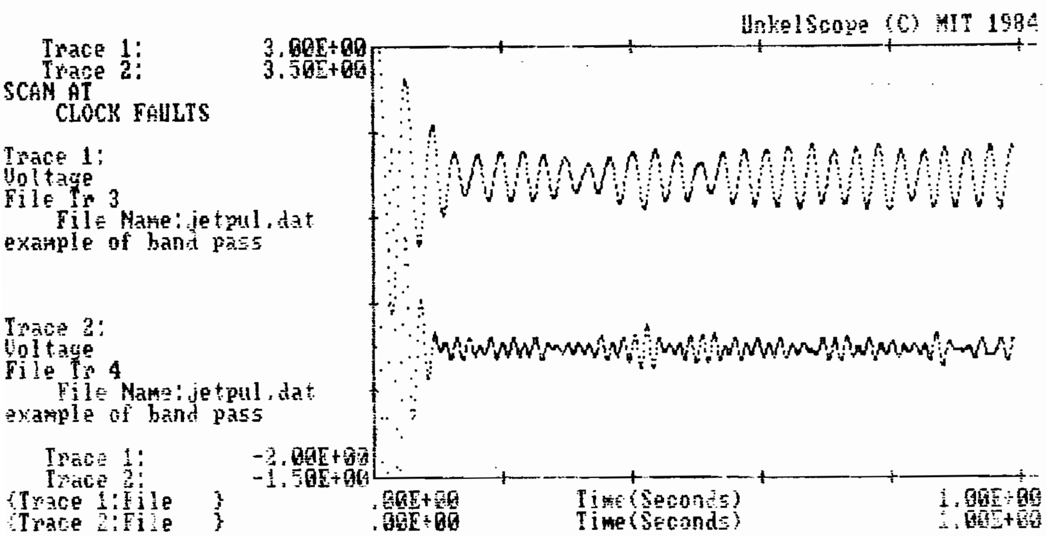


Figure 11.6c Effect of 2 band pass filters applied to the signal shown at the top of Figure 11.6b.

11.2 FFT Based Processing

The Fourier Transform provides a way to represent the frequency components of a single signal or the frequency or temporal relationship between two signals. The simplest representation that can be constructed with the aid of the Fourier Transform is the Power Spectral Density (PSD). The power spectral density shows what fraction of the signal strength lies within small frequency bands. For example, "white noise" has a uniform frequency content over the full range of frequency, and its PSD is a flat trace as shown schematically in Figure 11.7a. In contrast a pure sine wave has all its frequency component concentrated near the frequency of the sine wave and yields a PSD as shown in Figure 11.7b. Actual signals have both spikes and broad regions as indicated schematically in Figure 11.7c.

The digital or discrete form of the PSD has some peculiarities that need to be considered. First, the magnitude of the continuous PSD is represented only at particular frequencies resulting sometimes in a lack of resolving power. The increment in frequency between the computed elements is equal to $1/_{total\ sample\ time}$. Further, the results plotted for $f_{sample}/2$ to fsample are actually a "reflection" so that the actual useful range of the PSD is from f = 0 Hz to $f_{sampled}/2$. Finally, the data is processed as if the signal were truly periodic. This implies that the signal would repeat exactly scan after scan. This is usually not the case and a low frequency drift can cause a significant distortion of the signal. This last problem is partially removed by using "windows" as described later in this section.

Since the first frequency computed is 0, i.e. DC, it is often useful to remove the DC component of the signal. The software automatically removes the (average) DC level from the signal.

11.2.1 Applying the Power Spectral Density (PSD)

Move (using the left/right arrow keys) to the UTILITIES choice of the command mode and press the down arrow key to enter the Utility Command Mode. Toggle (using the left/right arrow keys) to the STATISTICAL choice and execute it by pressing the down arrow key. Note that the questions are sequential and there is no "editing" by moving back through the questions (i.e., the input is not handled in the same manner as the Setup Mode of the program).

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

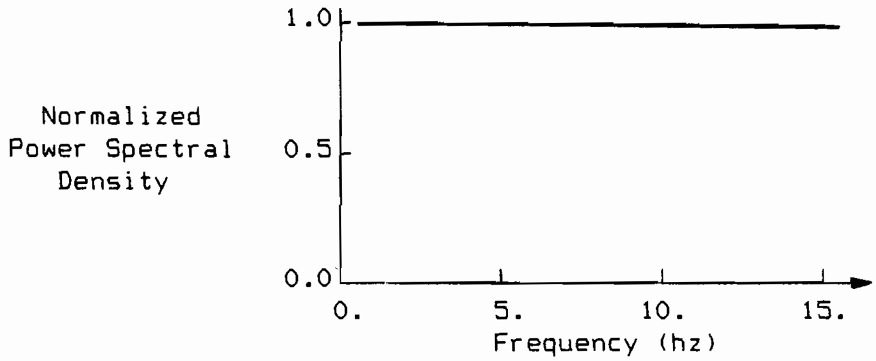


Figure 11.7a Power spectral density for 'white' noise.

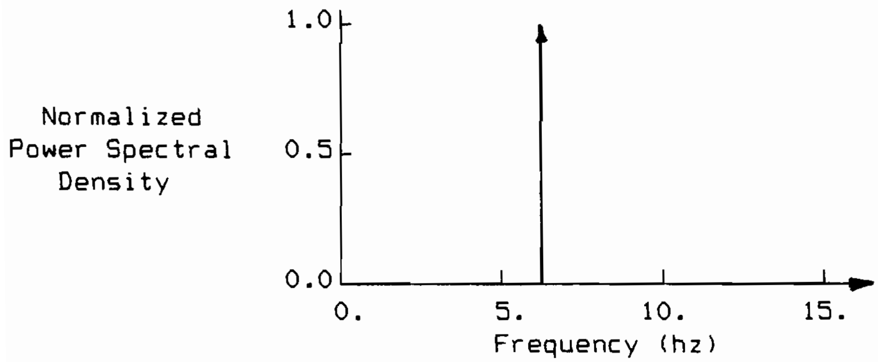


Figure 11.7b Power spectral density for a pure sine wave.

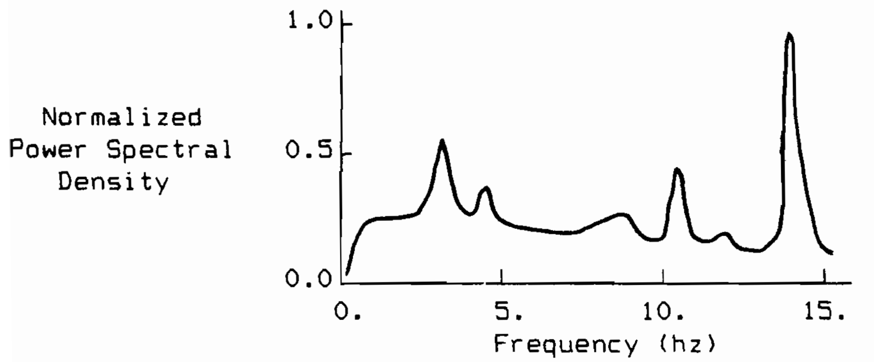


Figure 11.7c Power spectral density for a 'typical' signal.

The Inputs for FFT Related Functions

Input File Name: Enter the name of the file that contains the data to be used as the input. Use delete and Ctrl/U to fix mistakes. If the requested file does not exist, an error message will be printed in the upper left corner. You will be given an opportunity to examine the contents of the current directory and then an opportunity to try a different file name.

When the file is read, the label from the file is printed and some simple statistics about the file may be printed in the lower portion of the screen.

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

Selection is allowed sequentially for each valid channel. The program assumes the measurements to be spaced equally in time.

Operation. Use the left arrow key to toggle to a the Power Spectrum choice and then press the down arrow key.

Window. The window function w(t) is used to reduce the problem that the data is not truely periodic. The window function is applied to the original data y(t) so that

$$\underline{y}(t) = y(t) \ \ w(t)$$

and the transform is performed on the modified data $\underline{y}(t)$. UnkelScope allows you to select no windowing or one of three common window functions: the Hanning window, the Hamming window, and Top Hat window. These window functions are shown in Figures 11.8a through 11.8c. Use the right arrow key to select the window you desire.

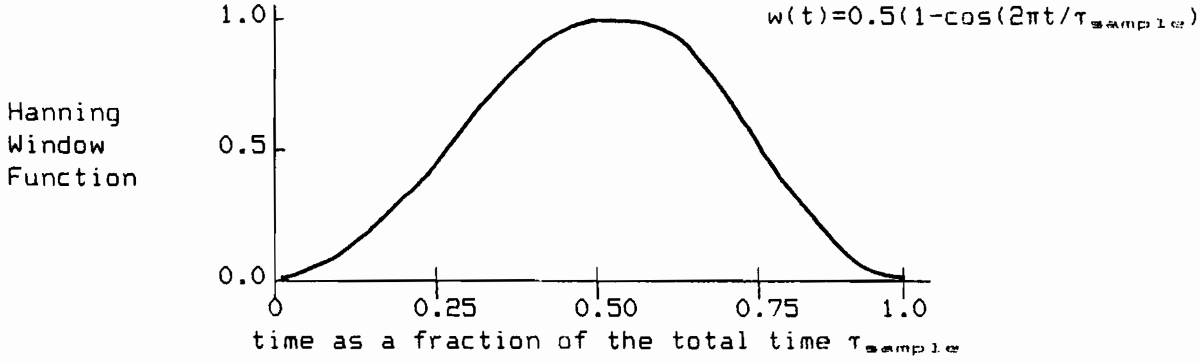


Figure 11.8a Hanning window function.

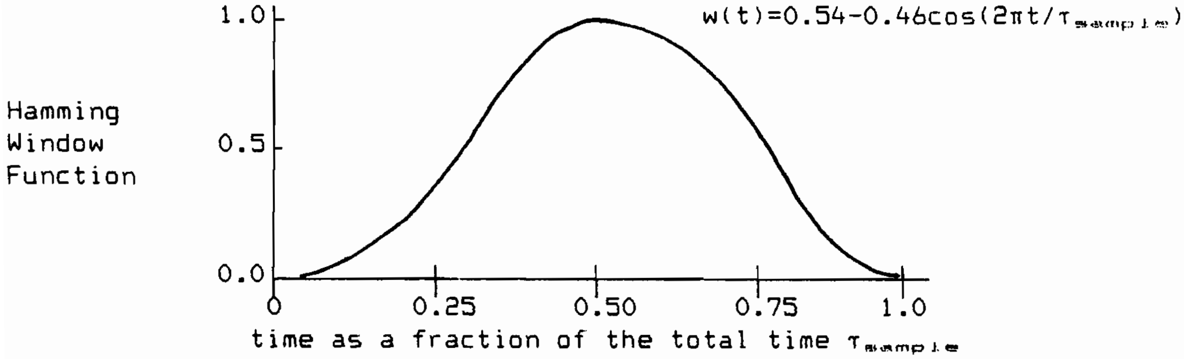


Figure 11.8b Hamming window function.

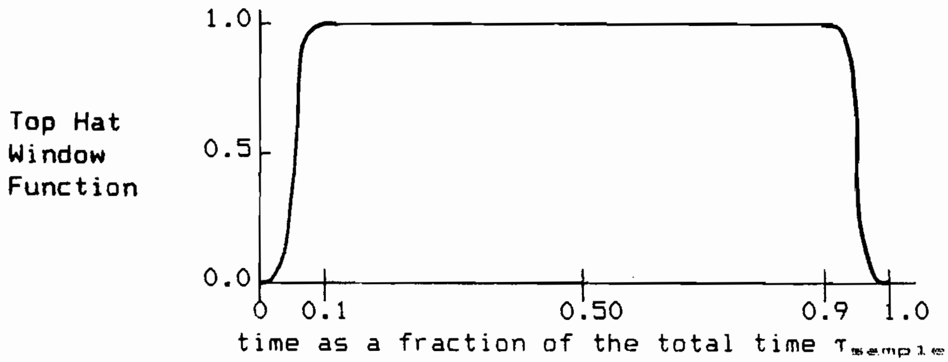


Figure 11.8b Top hat window function defined as:

$0 < t < 0.1\ \tau_{sample}$: $w(t) = 1-\cos(20\pi t/\tau_{sample})$

$0.1\ \tau_{sample} < t < 0.9\ \tau_{sample}$: $w(t) = 1$

$0.9\ \tau_{sample} < t < 1.0\ \tau_{sample}$: $w(t) = 1-\cos(20\pi t/\tau_{sample})$

Exiting and Obtaining Hardcopy. When the last signal has been processed, the program stores the file and prints the message

"Press Y to Return to Command Mode"

On the IBM PC you can get a hardcopy of the screen by typing Shift/PrtSc. Typing Y will return you to the command line of the Scope program with the cursor pointing to the UTILITIES choice.

11.2.2 Procedure to Display The PSD and Some Sample Results

The power spectrum results are stored with the magnitude in the vertical trace and the frequency in the time trace. The vertical trace block controls the scale of the PSD values, while the time base controls the range of the frequency display. The PSD is normalized to give a peak value of 1. The actual value of the peak is stored in the last 10 characters of the vertical trace label (overwriting what was already there.)

The range of frequencies of the PSD is from 0 to $f_{sample} = (1/t_{delay})$, but the frequencies from 1/2 f_{sample} to f_{sample} are, as discussed earlier, a reflection of the lower frequency results. Further, the frequencies are stored in kilohertz rather than Hz; the horizontal label is correctly constructed.

A set of data and its PSD with no window are shown as Figure 11.9a and 11.9b. Figure 11.9c shows the PSD of the same signal using a Hamming window.

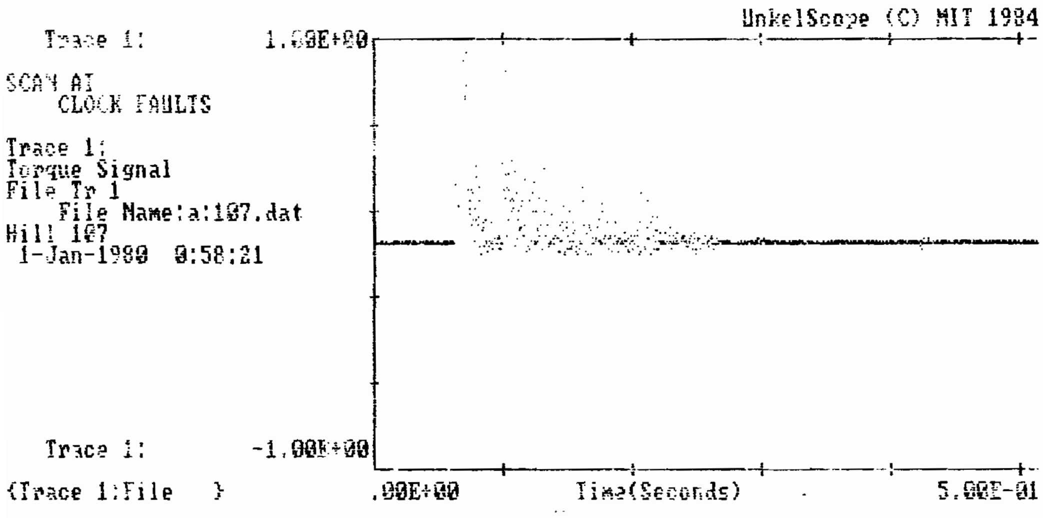


Figure 11.9a Plot of data used to obtain the results shown in Figure 11.9b and Figure 11.9c.

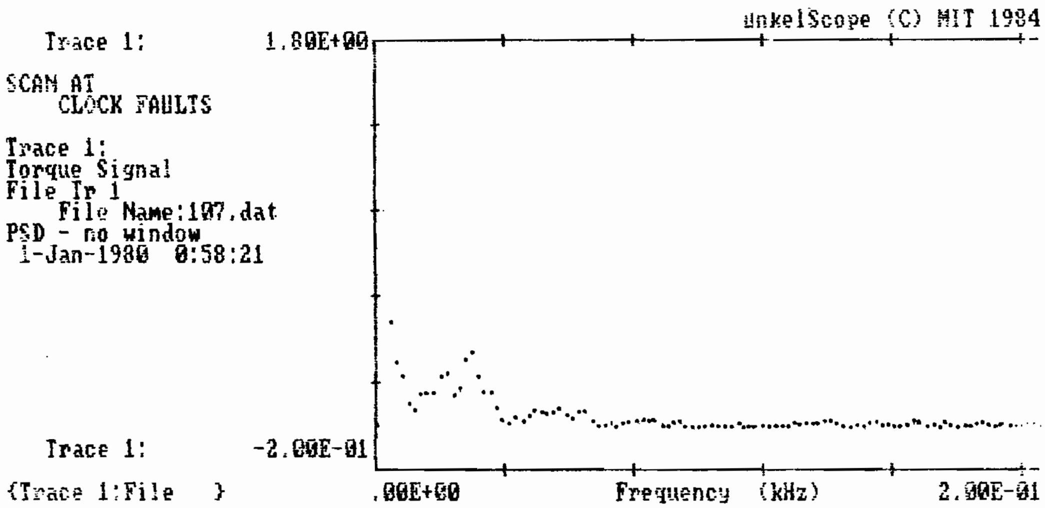


Figure 11.9b Plot of power spectral density of data shown in Figure 11.9a using no window function.

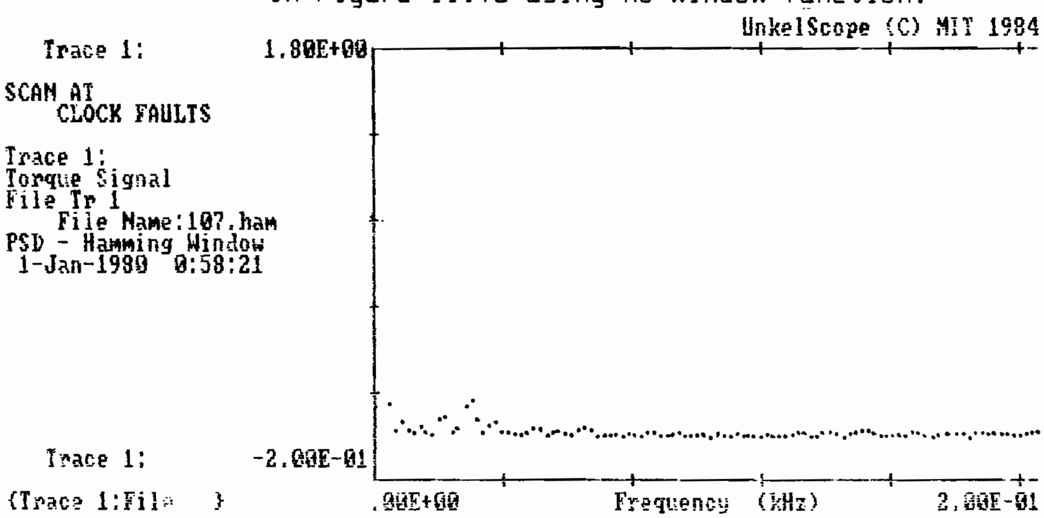


Figure 11.9c Plot of power spectral density of data shown in in Figure 11.9a using a Hamming Window.

12.0 Integration, Differentiation and Other Functions of Data

UnkelScope provides for manipulation of data as an out-of-line utility. A variety of functions are allowed and can be used to operate on data stored in files. Included as functions are integration and differentiation. The next section describes the general approach. The subsequent sections describe each function.

12.1 Procedure to Apply the Functions

Move (using the left/right arrow keys) to the UTILITIES choice of the command mode and press the down arrow key to enter the Utility Command Mode. Toggle (using the left/right arrow keys) to the FUNCTIONS choice and press the down arrow key to execute it. Note that the questions are sequential and there is no "editing" by moving back up the menu (i.e. the input is not handled in the same way as the setup mode of the program).

Entering Numbers. The numbers are entered as up to 9 characters in "i", "f" or "e" FORTRAN type format. For example, the number 92 could be entered at 9.2E+1 or 92. or 92 or 920.E-1. Pressing the return key, typing the 10th character or typing a comma signals the end of a number. During the entering of a number, the delete key deletes the last character and typing Ctrl/U deletes all the characters describing that number. If the program cannot interpret your input an error message is printed in the upper left of the screen.

Entering Labels (Alpha Numeric.) Labels are entered by typing the characters. The delete key deletes the last character and Ctrl/U deletes all the characters in the label entry. The label is ended when you press return, the down arrow key or when you type a character beyond the length expected.

Inputs to The Function Utility

Input File Name: Enter the name of the file that contains the data to be used as the input. Use delete and Ctrl/U to fix mistakes. If the requested file does not exist, an error message will be printed in the upper left corner. You will be given an opportunity to examine the contents of the current directory and then an opportunity to try a different file name.

When the file is read, the label from the file is printed and some simple statistics about the file may be printed in the lower portion of the screen.

Output File Name: Enter the name of the file in which the modified data will be stored. If this name is specified to be the same as the input file name, a message is printed in the upper left corner and you will be given a chance to modify it so it is not the same as the input file. Use Delete and Ctrl/U to fix mistakes.

Label for Output File: The label from the input file will be copied here and you will be allowed to modify it. Use Delete and Ctrl/U to make the alterations.

Before allowing the options to be selected, the input file is read. If the input file does not exist, or is not a valid data file, an error message is printed in the message line.

Selection is allowed sequentially for each channel of valid data. Operations can be performed only on the vertical trace and the horizontal trace (i.e. not time). As each new signal is ready to be processed, some simple statistics (minimum, maximum and average) of that signal are printed.

Function. After you have accepted the label for the trace you can toggle to choose which function to be performed.

Exiting and Getting Hardcopy. When the last signal has been processed, the program stores the file and prints the message: "Press Y to Return to Command Mode." On the IBM PC you can get a hardcopy of the screen by typing Shift/PrtSc. Type Y to return to the command line.

12.2 Example of Integration of a Signal

Figure 12.1 shows a signal and its integral indicating the "smoothing" of the data that accompanies integration.

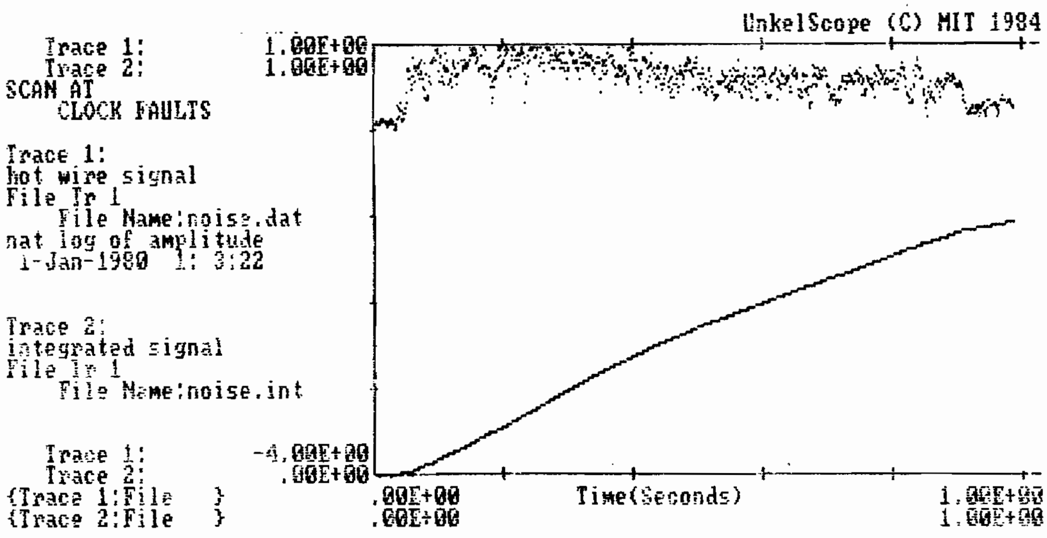


Figure 12.1 Plot of a signal and its integral.

12.3 Examples of Differentiating a Signal

Figure 12.2a shows a signal and its derivative, indicating the intensification of the noise in the signal. Figure 12.2b shows the results obtained when the signal is filtered (with a low pass filter of cutoff 10 Hz) and then differentiated.

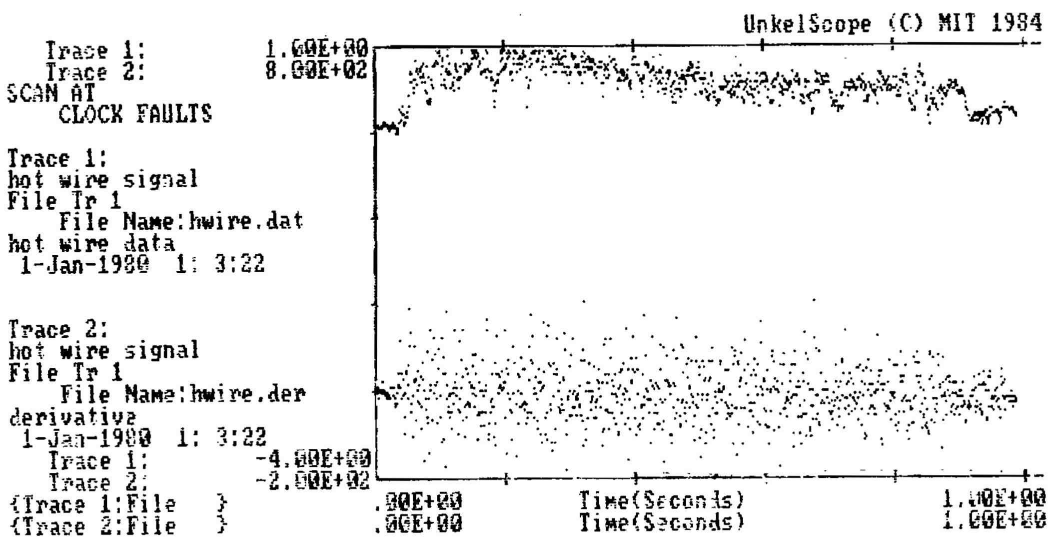


Figure 12.2a Signal and its derivative, showing how the noise in the signal is intensified by the process of differentiation.

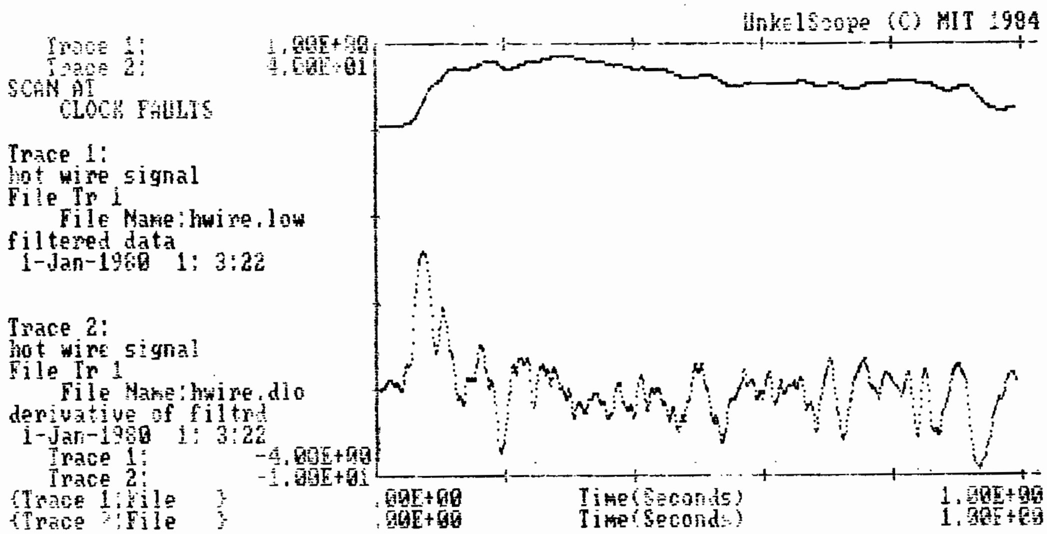


Figure 12.2b Signal of Figure 12.2a filtered and then differentiated. Note the reduction in the noise level.

12.4 Other Functions

A variety of other functions are provided and are discussed below. In particular, it is important to know what is done in the case of errors such as trying to divide by zero. Section 12.5 discusses the use of the functions for one particular example.

12.4.1 Subtract DC Value

The program finds the average value over all data points and subtracts this value from each data value.

12.4.2 Absolute Value

The program takes the absolute value; this behaves like an analog full wave rectifier.

12.4.3 Peaks from Start

This function sets the signal at a time "t" equal to the absolute maximum of the signal in the interval from the first sampled time to the current time to the last sampled time. Figure 12.3a shows an example of the effect of this function.

12.4.4 Peaks from End

This function sets the signal at a time "t" equal to the absolute maximum of the signal in the interval from the last sampled time to the current time. Figure 12.3b shows an example of the effect of this function.

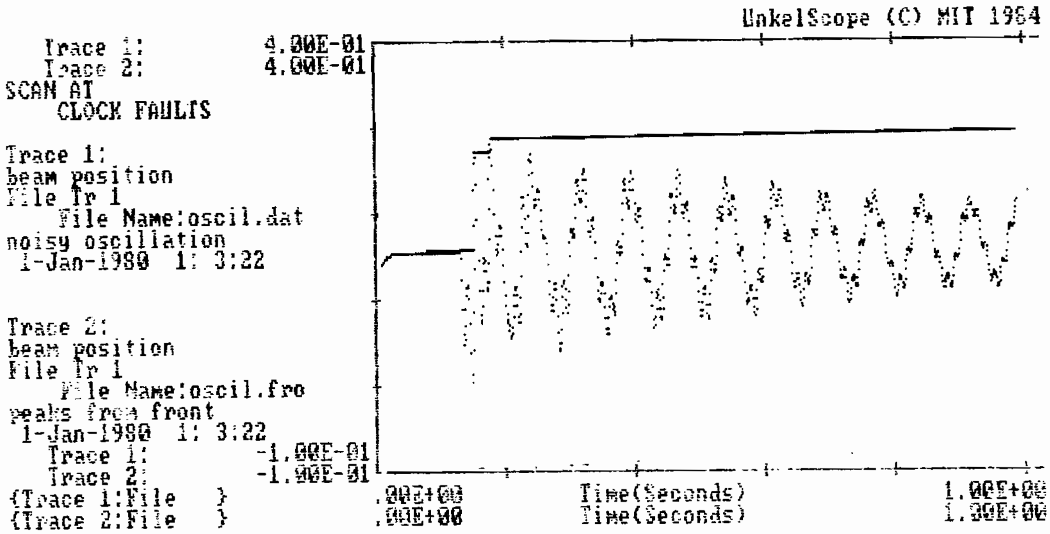


Figure 12.3a Signal and result of applying the
'peaks from start' function.

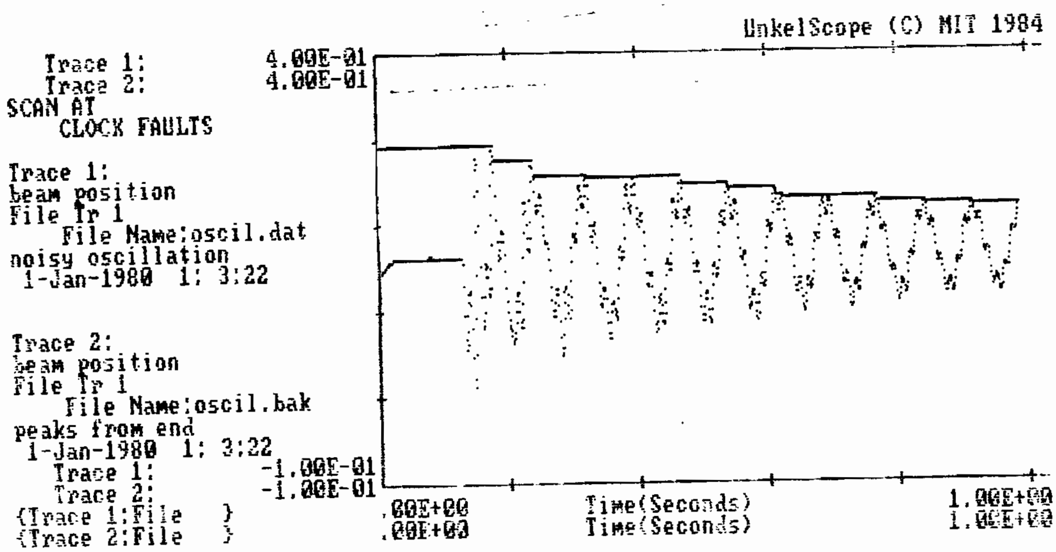


Figure 12.3b Signal and result of applying the
'peaks from end' function.

12.4.5 Floor(minimum)

This applies a minimum value, such that any data value below the minimum is set equal to the the minimum value.

12.4.6 Ceiling(maximum)

This apples a maximum value, such that any data value above the maximum is set equal to the maximum value.

12.4.7 Natural Log

This function takes the natural log of each value. For any negative values the output values are set to -1.E22.

12.4.8 Exponential

This function takes the exponential of each data point. For any inputs greater than 50 the output is set to 1.E22; for input data less than -50 the output is set to 1.E-22.

12.4.9 Square

This function squares each data value.

12.4.10 Square Root

This function finds the square root of each data value. For each input less than zero, the output value is set to -1.E22.

12.4.11 Inverse

This function takes the inverse of the data. For each 0.0 value, the output value is set to 1.E22.

12.5 Example of Use of Functions : Identification of The Parameters of a Second-Order System

An oscillating second-order system shows the characteristic decay shown schematically in Figure 12.4a. The frequency of oscillation can be established by counting the oscillations in a known time; or it can be found by using the Power Spectral Density (PSD) as indicated by the results shown in Figure 12.4b. (See Section 11.2 for a full description of the PSD function.)

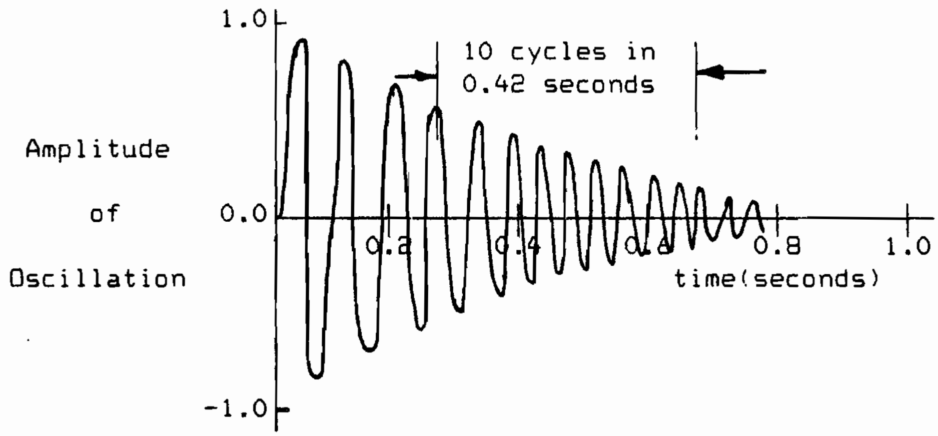


Figure 12.4a Decay of an underdamped second-order system to an impulse.

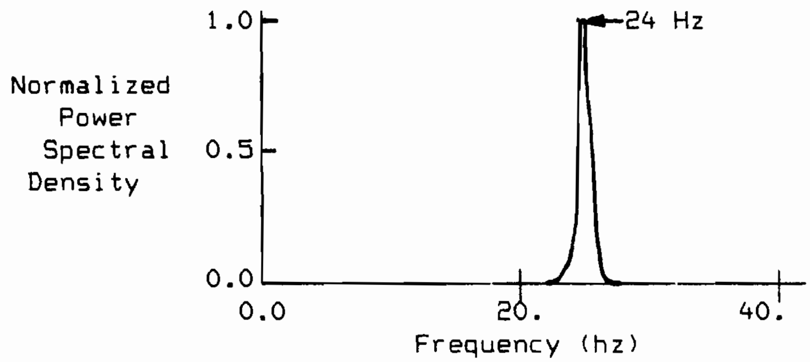


Figure 12.4b Power spectral density for the data shown in Figure 12.4a.

The parameter characterizing the damping can be found from the envelope of the decay. Specifically, the envelope should follow the relation

$$V_{out}/V_{out}(t_o) = \exp [- z\ w_n\ (t - t_o)]$$

where z is the damping ratio, t_o is a reference time and w_n is the natural frequency. Taking the natural log of both sides gives

$$\ln (V_{out}/V_{out}(t_o)) = - z\ w_n\ (t-t_o)$$

Thus for a linear second order, underdamped system the slope of the ln (amplitude) vs. time should be a straight line and the slope of the line gives the value of z. The Situation is shown in Figure 12.5

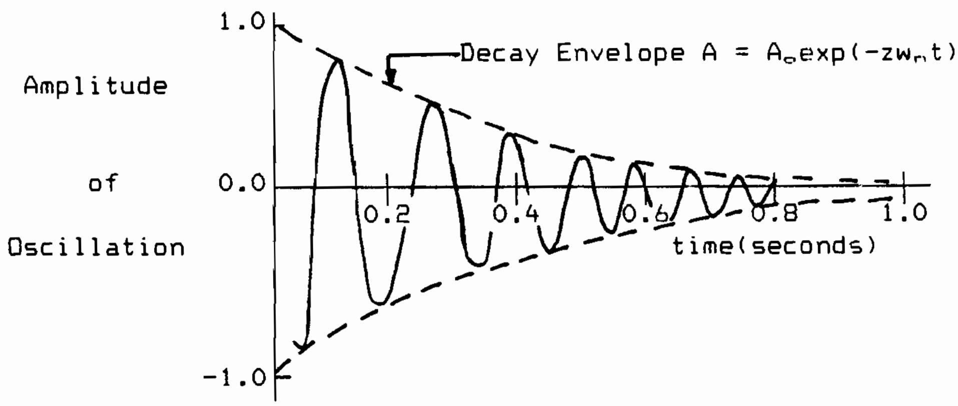


Figure 12.5 Decay of the system and identification of the envelope in terms of the damping coefficient.

The functions in UnkelScope Level 2 can be combined to process data as shown in Figure 12.6 resulting in a display of the envelope in the form necessary to get the damping ratio. The sequence of steps and the results of each step are shown below and in Figures 12.7a through 12.7f.

Step A: Filter the Data. The signal of interest must be near the natural frequency so the signal can be low pass filtered to remove any high frequency noise apparent in the measured signal.

Step B: Remove Any DC Signal. The amplifier introduced a DC offset that must be removed before the natural log is taken, because the amplitude is the peak relative to zero.

Step C: Take The Absolute Value. This will double the number of peaks in the interval and improve the overall character of the envelope.

Step D: Detect Peaks From End. This will form a representation of the envelope as a "staircase".

Step E: Take The Natural Log. This results in a straight line, at least for times where the noise in the signal is large relative to the signal.

Using the final plot, the slope can be determined and the value of z found.

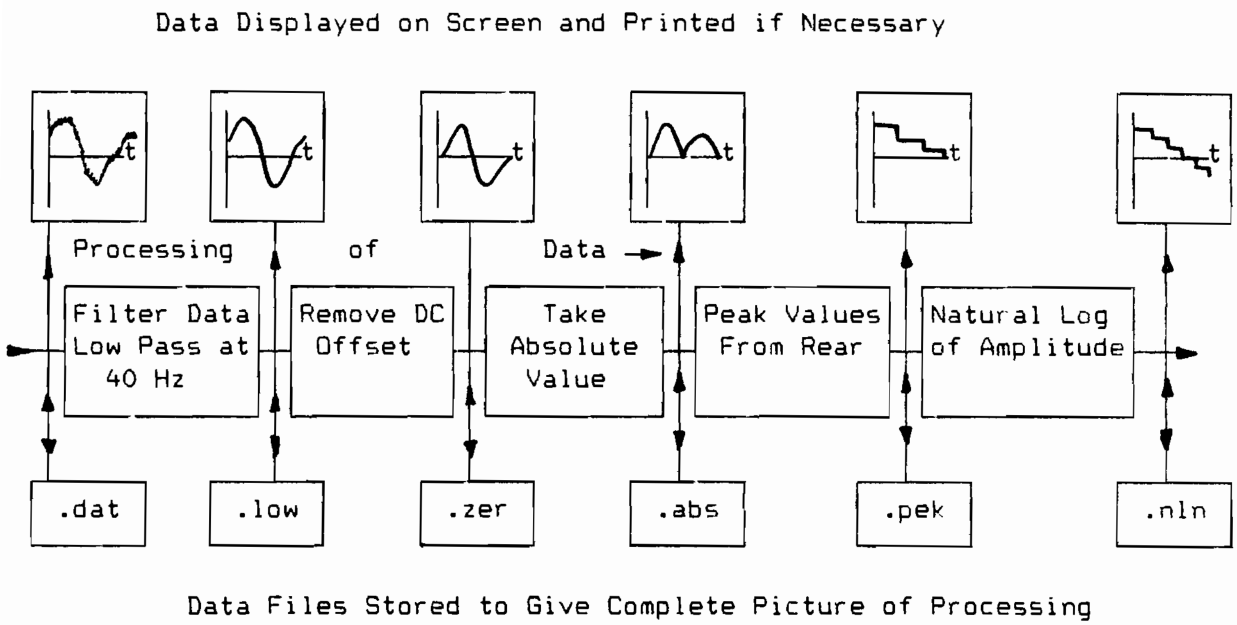


Figure 12.6 Schematic of data processing of response data to determine the damping ratio.

12.6 Summary

The Function Utility allows the vertical or horizontal trace data to be modified by a variety of functions. These functions can be applied sequentially to conveniently process a data signal from a 'raw' form to the desired results.

Passing Data from Utility to Utility without writing Disk Files

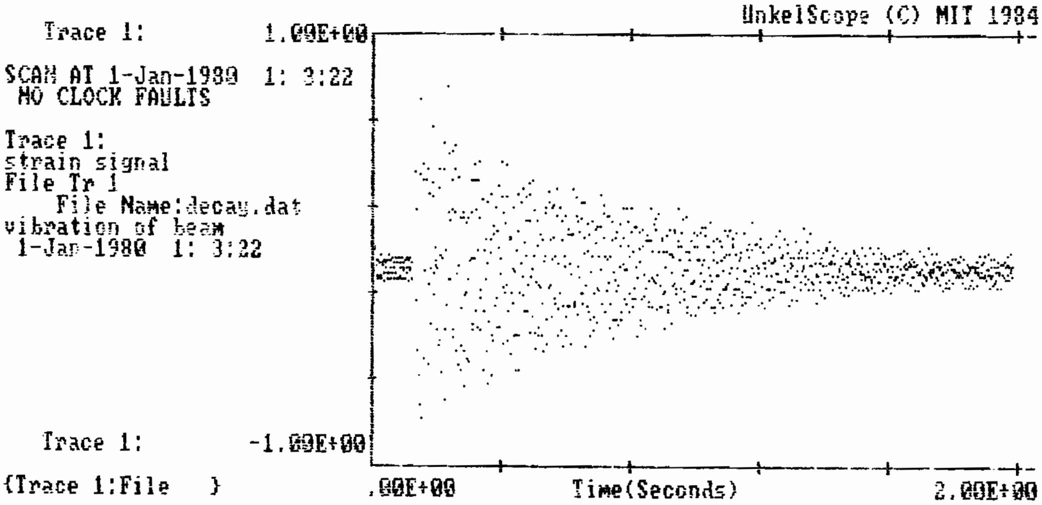
Under some circumstances you may not wish to see the intermediate results of calculations done by the Utilities. If this is the case, you can have UnkelScope pass the information from step to step provided that you:

a) do not leave the Utilities in between steps, and

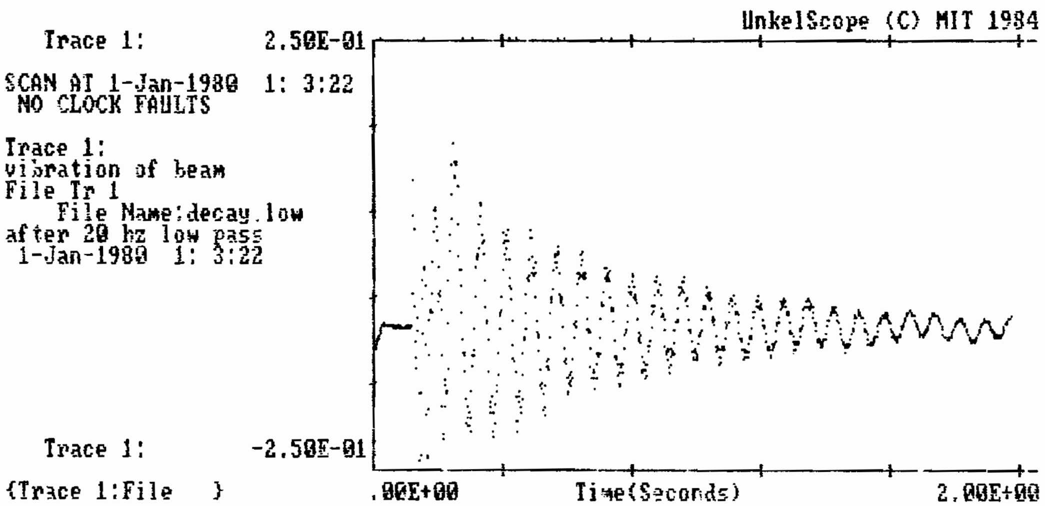
b) you store the last step (and some of the others along the way.)

To use this feature specify a file name of = (the character =) as the output of one step and as the input of the next step.

The reason that you cannot exit the Utility section is because UnkelScope shares the storage for the Utilities with some of the storage for sampled data. If you try to exit the Utilities when the last output file was specified as "=", UnkelScope will give you a chance to go back and save this data (simply execute one of the utilities with no changes made except for specifying an output file name.) If you do not save the data at this time, the data will be destroyed and you will not be able to save it later.

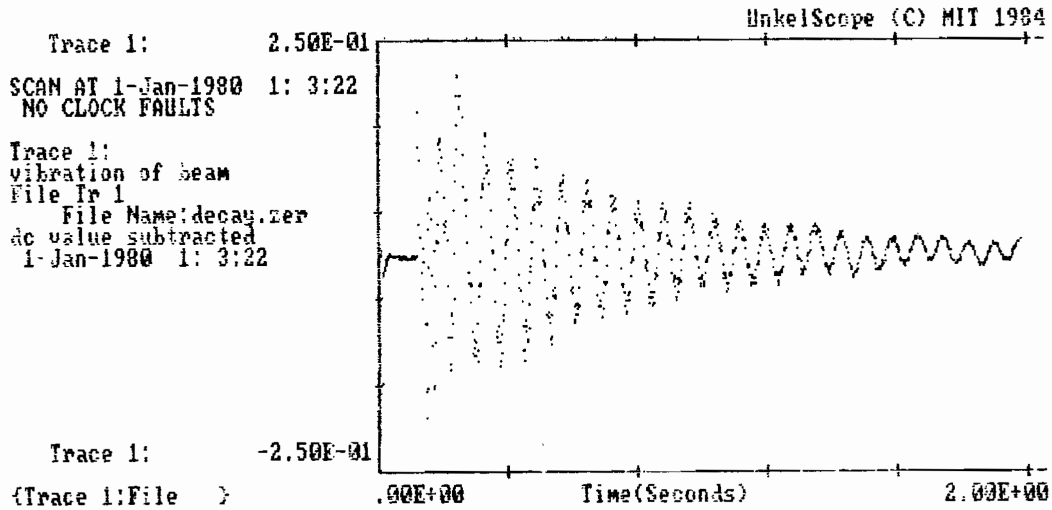


Raw data

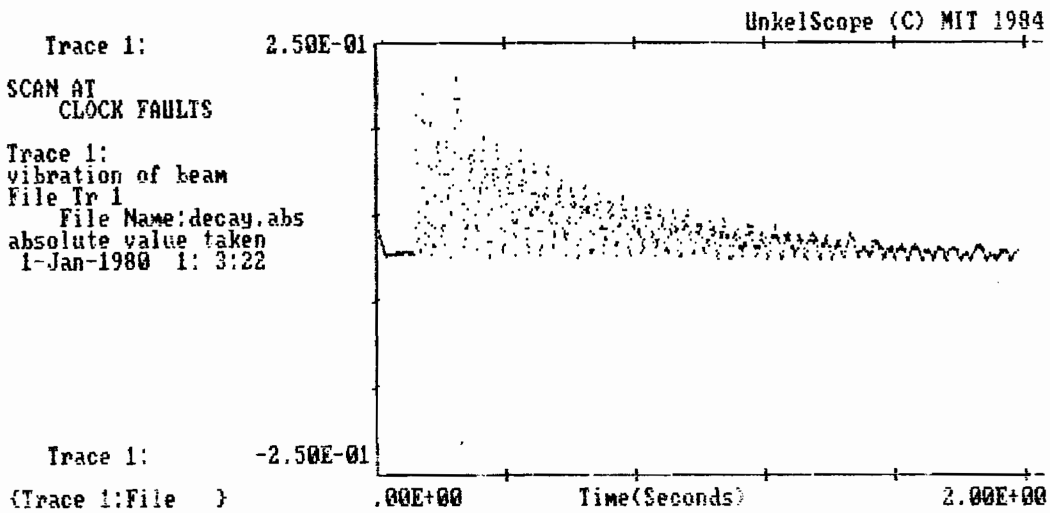


Data after smoothing by a low pass filter

Figures 12.7a-12.7f Progression of the signal shown in Figure 12.7a as it is processed to obtain the plot that shows the decay constant. (Page 1 of 3.)

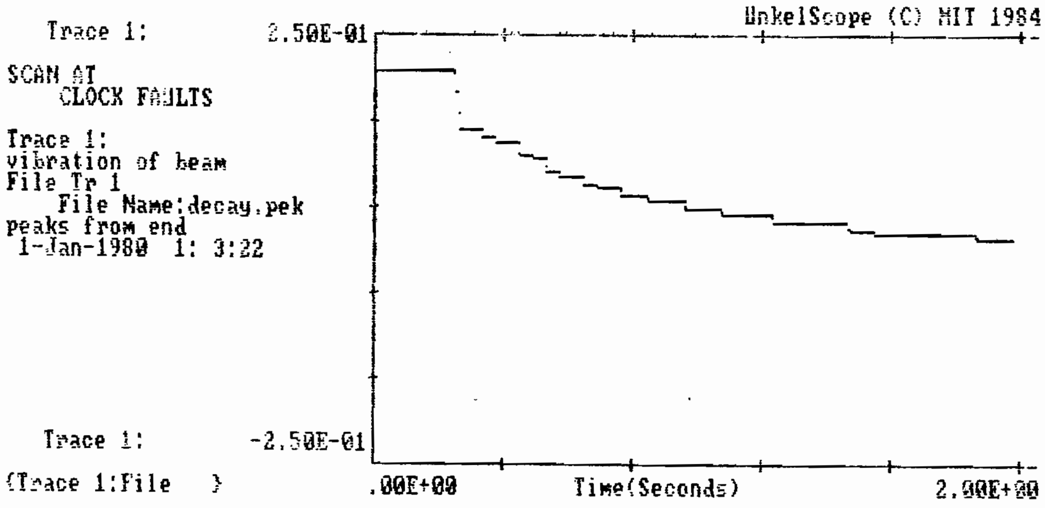


Data After Average Value is Subtracted

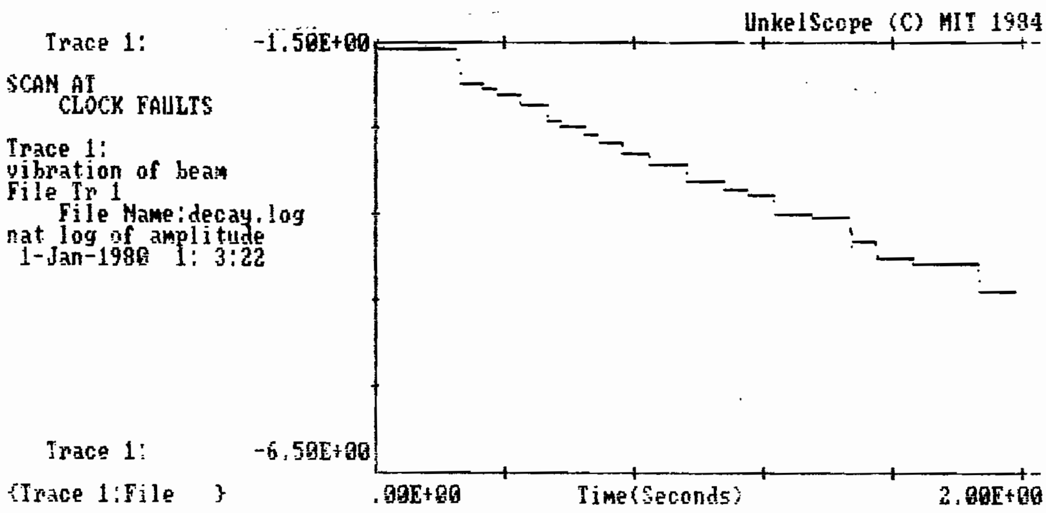


Data after absolute value is taken

Figures 12.7a-12.7f Progression of the signal shown in Figure 12.7a as it is processed to obtain the plot that shows the decay constant. (Page 2 of 3.)



Data after peaks from end has identified the decay envelope



Data represented as the log amplitude of the decay envelope

Figures 12.7a-12.7f Progression of the signal shown in Figure 12.7a as it is processed to obtain the plot that shows the decay constant. (Page 3 of 3.)

13.0 Experiment Control, Controllers and Coordinated Stepping and Sampling

UnkelScope allows for three types of experiment manipulation:

a) control of devices using the digital output port;

b) open loop and proportional controllers; and,

c) sampling coordinated with traversing or stepping.

These functions are provided only for the slower sampling rates and only one type of control function can be exercised at a time. In the sampling block of the setup menu, the message (Processing Active) notifies you that one of these functions can be used. Each control function is discussed separately in the three sections following. These functions are activated by moving to the User Processing Line of the Setup Mode and toggling to the type of control. The choices are none, relay control, controller (more than one type), traversing; also under User Processing is the on-line linear conversion option described in Section 14.

13.1 Control of Devices With Digital Output Port

Many times it is necessary to control some external events by switching them on or off during the course of the experiment or it is necessary to protect against a dangerous condition by shutting down a device when a critical level is achieved. UnkelScope uses 4 lines (bits) of the digital output port to indirectly or directly control 4 devices. Operation is divided into 5 "states" with each having its own combination of the "on"/"off" state of the 4 digital lines. Operation proceeds sequentially from one state to the next with the transition from one state to an other effected when a trigger condition is met. The trigger condition can be based on time or on the signal measured as one of the Vertical Traces. The operation is shown schematically in Figure 13.1. For triggering from the Vertical Trace inputs, the slope and level are specified. As shown in Figure 13.1 when the signal crosses specified level with the defined slope the transition occurs. For a time activated transition, only positive slopes, i.e., increasing times are allowed and for each new scan the time value is reset to zero.

Condition For Change to Next State		Digital Line States Out 1	Out 2	Out 3	Out 4
System Put In State A at Sampling Start	State A	on	on	off	off
Vert Trace 1 Increases Above 3 Volts	State B	off	off	off	on
Vert Trace 1 Decreases Below 3 Volts	State C	off	off	off	off
Vert Trace 1 Decreases Below -2 Volts	State D	on	off	off	on
Time Increases Past 49 Seconds	State E	on	on	off	off

Figure 13.1 Schematic of digital output control showing the "states" and the triggering from one state to the next.

The software interface is shown in Figure 13.2. Note that the down and up arrow keys move to next and previous choices within the same line. The left and right arrow keys alter the current option, e.g., change on to off. The triggering levels are entered as numbers rather than being toggled.

<u>Entering Numbers</u>. The numbers are entered as up to 9 characters in either "i", "f" or "e" FORTRAN type format. For example the number 92 could be entered at 9.2E+1 or 92. or 92 or 920E-1. The return key (or typing the 10th character) signals the end of a number. When a number is being entered, the Delete key deletes the last character and Ctrl/U deletes all the characters describing that number. The program tries to interpret numbers with peculiar characters but may not be able to. If the number is not in the correct form a message is printed to the messsage line.

```
------------ Vertical Trace 1 -----------   -------- Additional Vertical Traces -------
Source [Analog  0]  A/D Range [+/- 10]      Tr Input                              A/D
Label  :Critical Voltage                    #  Chan  Label:                     Range
Span   [ 10    v full scale]                3 [  1  ]Voltage                 [+/- 10]
Range  [-5.00E+00 to  5.00E+00]             4 [none ]

------------ Vertical Trace 2 -----------
Source [None      ]

                                            ------------ Triggering ------------
                                            Mode  [Singl Sweep] Source [Keyboard ]

------------ Horizontal Trace -----------
Source [Time      ]                         ------------ Processing ------------
Label  :Time(Seconds)                       Type [Relay Control            ]
Span   [50    s full scale]                 St Src  Slp    Lev:     1   2   3   4
Range  [  .00E+00 to  5.00E+01]             A  Time [+]  .00E+00   on  on  off off
                                            B  Tr 1 [+] 3.00E+00   off off off on
--------------- Sampling ----------------   C  Tr 1 [-] 3.00E+00   off off off off
Sample Rate [ 50 ms          20 hz]         D  Tr 1 [-]-2.00E+00   on  off off on
[1024] Samples  (Scan Time 5.12E+01 s)      E  Time [+] 4.90E+01   on  on  off off
(Real Time Plot) (Processing Active  )
```

Figure 13.2 Relay control interface showing usage of time and vertical trace initiated transitions.

As an example consider a system where a pump is turned on after a fixed time and used to fill a tank to a specified level. The system is shown schematically in Figure 13.3a. The pump motor can be switched on by a solid state device that accepts a TTL logic input from the Digital Output Line 1. A pressure transducer measures the gage pressure at the bottom of the tank, from which the height of water can be determined. At the desired height of 10" of water, the transducer output 3.5 volts. Figure 13.3b shows the setup configuration for this example; Vertical Trace 1 measures the pressure signal while Vertical Trace 2 monitors the digital output 1 which controls the motor. The motor is turned on after 5 seconds and turned off when the voltage measured as Vertical Trace 1 increases (positive slope) above the value 3.5 volts. The resulting response of the water height (as measured by pressure at the bottom of the tank) and the control of the pump motor are shown in Figure 13c. As can be seen the tank filled to the desired height about 20 seconds after the pump was switched on.

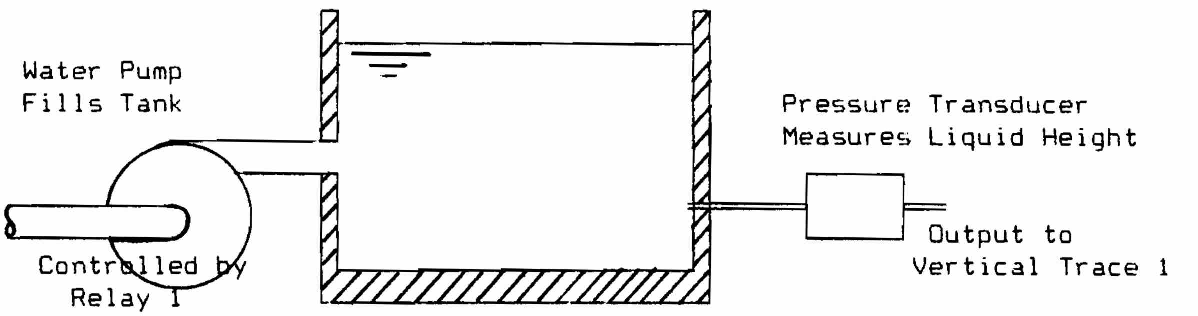


Figure 13.3a Schematic of equipment used in the Example of Section 13.1

```
-------- Vertical Trace 1 --------          ------ Additional Vertical Traces ------
Source [Analog  0]  A/D Range [+/- 10]      Tr Input                           A/D
Label  :Pressure Signal                     #  Chan  Label:                  Range
Span   [  5    v full scale]                3 [none ]
Range  [  .00E+00 to  5.00E+00]             4 [none ]

-------- Vertical Trace 2 --------
Source [None     ]

                                            ------------ Triggering ------------
                                            Mode  [Singl Sweep] Source [Keyboard ]

-------- Horizontal Trace --------
Source [Time     ]                          ------------ Processing ------------
Label  :Time(Seconds)                       Type [Relay Control           ]
Span   [50   s full scale]                  St Src  Slp    Lev:     1   2   3   4
Range  [  .00E+00 to  5.00E+01]             A  Time [+]  .00E+00  off off off off
                                            B  Time [+] 5.00E+00  on  off off off
------------ Sampling ------------          C  Tr 1 [+] 3.50E+00  off off off off
Sample Rate [ 50 ms        20 hz]           D  Time [+] 4.70E+01  off off off off
[1024] Samples  (Scan Time 5.12E+01 s)      E  Time [+] 4.90E+01  off off off off
(Real Time Plot) (Processing Active  )
```

Figure 13.3b Selections of setup mode for the example of Section 13.1.

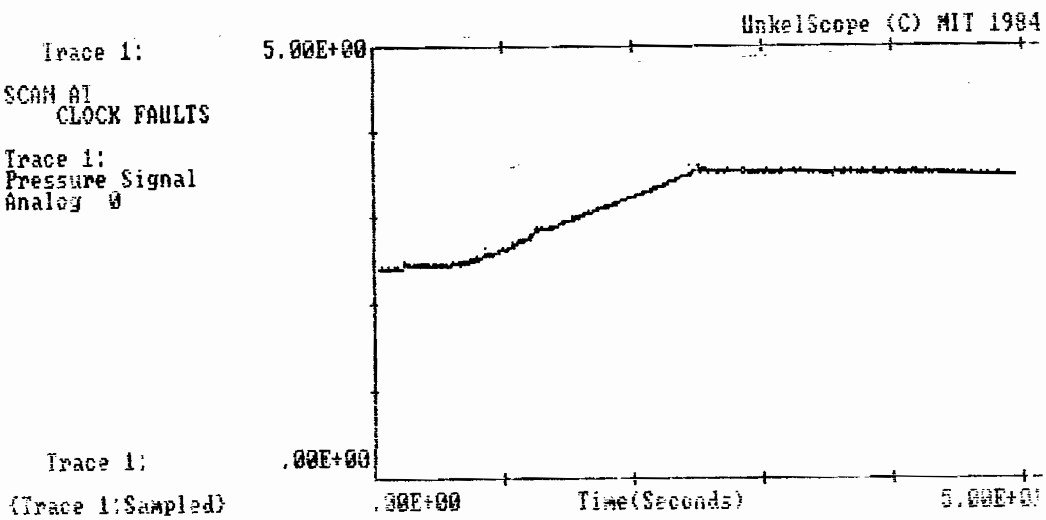


Figure 13.3c Variation of the water height as the tank is filled.

13.2 Process Controllers

The computer can be used as a part of a digital control system with the digital-to-analog (D/A) converter providing the output signal in response to a command signal with or without comparison to a feedback signal. This feature is not available for boards without a digital output channel; this option will not appear in the menu if your hardware does not support it.

The present software allows for a few types of process controllers and gives full choice of the signal sources. The software interface shown in Figure 13.4 matches a conventional sketch of a controller. The command signal is drawn from any of the Vertical Traces, which means it can be derived from a file of previously stored data. The feedback signal can also be taken from any of the Vertical Traces. The output signal can be sent to any valid D/A in the system. There are several controller options from the simplest open-loop system to a proportional-integral- derivative controller. Each type of controller requires a different set of inputs.

The controllers work only in the slow speed mode. In the setup menu a message is printed in the lower left corner that indicates whether the controller option is in effect. The message reads {Processing Active} or {Processing Inactive}. This is not a choice you have, but depends upon the hardware you are running (both the computer and the board.) Also note that for some boards, special hardware connections may be required; if this is the case the Installation Guide and the Short Reference Guide will detail these special connections.

The program truncates values that would cause output values outside the range of the hardware, but there is no indication that this has been done.

The open-loop controller can also be used as a function generator; the desired functional variation is stored in a data file and that data file is used as the command signal output to the digital-to-analog converter.

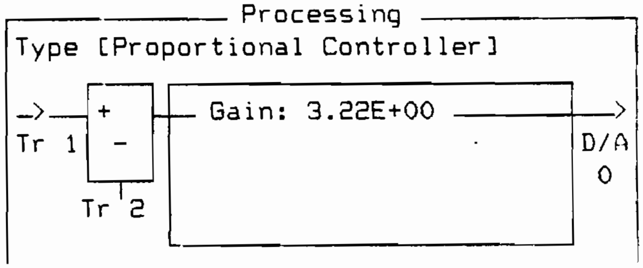


Figure 13.4 Software interface for open loop or proportional controllers.

Figure 13.5 shows application of the controller software to a speed controller. A servo motor powering a load is driven by the (power amplified) output of the D/A converter. A tachometer is used to measure the rotational speed and a proportional controller applied to the system. Figures 13.6a and 13.6b show the step response and ramp response for controllers with different gain settings chosen. The results show the expected improvement in steady-state response and rise time as the gain is increased.

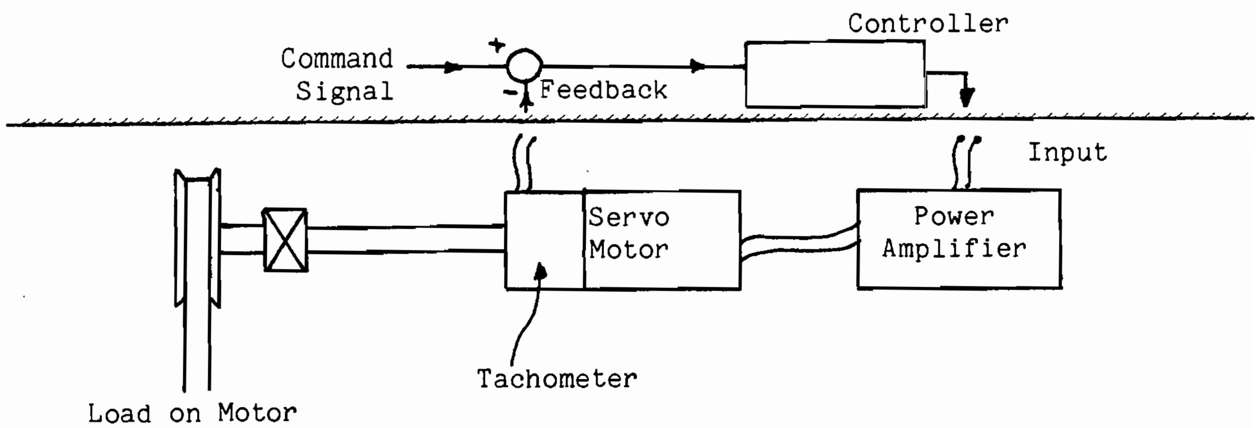


Figure 13.5 Hardware setup for speed controller.

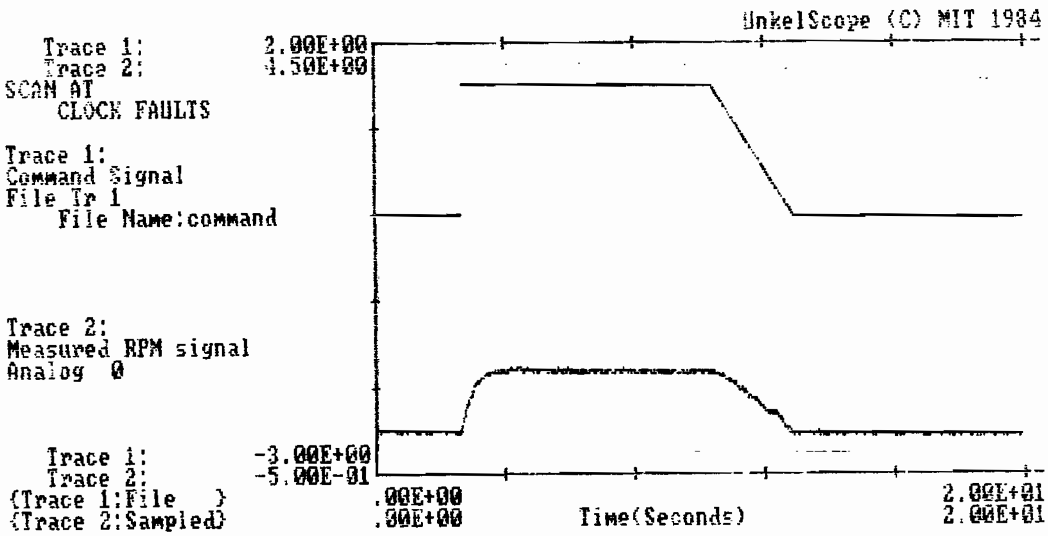


Figure 13.6a Command and measured response of the speed controller with a proportional gain of 2.

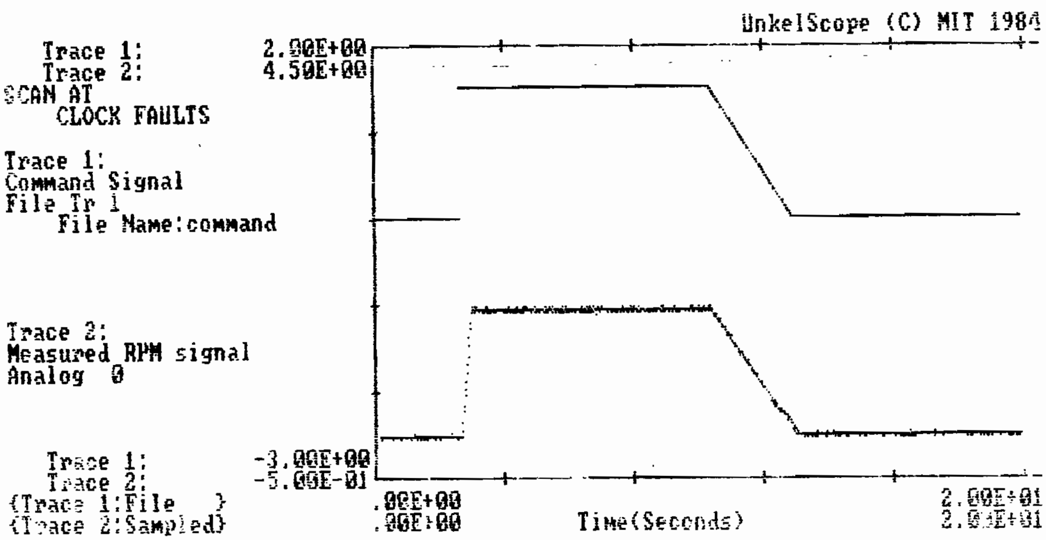


Figure 13.6b Command and measured response of the speed controller with a proportional gain of 20.

<u>PID Controller</u>. A PID Controller is also available with UnkelScope. There are many ways to implement the PID Control scheme, and UnkelScope uses the simplest of these for its controller. The classical PID algorithm has the form

$$\text{output}(t) = G_p\,\text{error}(t) + G_i \int \text{error}(t)\,dt + G_d \frac{d}{dt}\,\text{error}(t)$$

where output(t) is the control or output signal at time t, error(t) is the difference between the command signal and the feedback signal and G_p, G_i, and G_d are the proportional, integral, and derivative gains.

A straightforward discretization of this equation gives, in the z-domain,

$$U(z^{-1}) = (G_p + h\,G_i/(1-z^{-1}) + G_d\,(1-z^{-1})\,h)\,E(z^{-1})$$

where U is the output, E is the error and h is the time step of sampling. The expression can be expanded and terms rearranged, and the expression stated in difference equation form as

$$u(t) = u(t-1) + b_o\,e(t) + b_1\,e(t-1) + b_2\,e(t-2)$$

where

$$b_o = G_p + h\,G_i + G_d/h$$

$$b_1 = -(G_p + 2\,G_p/h)$$

$$b_2 = G_d/h$$

In the UnkelScope implementation, the computed value of the output is limited to be between 10^5 and -10^5; the actual output from the D/A converter is further limited to the hardware maximums. Thus, this controller does not have "anti-windup" control. Also, the derivative term is unfiltered.

The figure below shows the PID controller section, and how the three gains are identified. The sampling time is taken from the value in the Sampling Block.

```
 ------------- Horizontal Trace -------------
| Source [Time      ]                        |
| Label  :Time (seconds)                     |
| Span   [50   s full scale]                 |
| Range  [  .00E+00 to  5.00E+01]            |
 --------------------------------------------
 ---------------- Sampling ------------------
| Sample Rate [ 50 ms          20 hz]        |
| [1024] Samples  {Scan Time 5.12E+01 s}     |
| {Real Time Plot} {Processing Active  }     |
```

```
 ------------------ Processing -------------------
| Type [PID              Controller]              |
|                                                 |
| -->--| + |--|-Prop : 1.00E+01---------|---->    |
| Tr 1 | - |  |-Integ: 1.00E+00         |  D/A    |
|        |    |-Deriv: 1.00E+00         |   0     |
|      Tr 2   |___________________________|         |
```

Initial and Final Values. UnkelScope sets the D/A to zero when you enter the SMPL/DSPLY mode and resets the value to zero when you return from the SMPL/DSPLY mode. For the PID Controller, the value at the end of the each scan is kept until either a new scan is initiated, or until the program exits the sampling mode. You must make sure that these actions do not cause damage or disruption of your apparatus. Note that with the Open Loop and Proportional Controllers, the D/A output is reset to zero at the end of each scan.

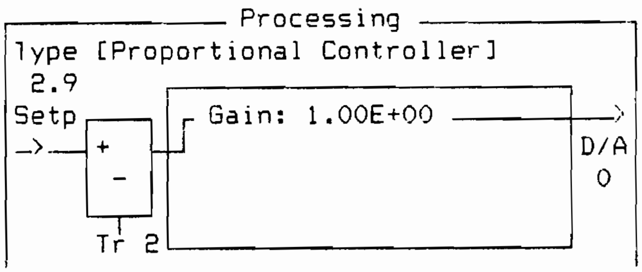
A reference summarizing PID controllers is : Clark, D. W., "PID Algorithms and their Computer Implementation," Trans. Inst. M C, Vol. 6, No. 6, Oct-Dec, 1984.

Control Features added in Release 2.25 and Later

The new control features have been added to the capabilities of UnkelScope in Release 2.25 are described here. All operations previously available are still available.

1. Setpoint Value for Command Signal

In addition to the Trace Source for the Command Signal, it is also possible to set a constant value "Setp" for the command signal. In the setup menu, toggle to the Setp option, press the down arrow and then enter the value shown above the label as shown below. Remember: to change the value, first backspace over the characters and then retype them.



Controller Block of the Setup Menu Showing the "SetP" option.

2. Alteration of Controller attributes during operation.

You can now interrupt a scan, change some of the controller parameters and then continue. While the sampling is interrupted, the control signal (output) is held constant.

To alter values during operation:

a. Interrupt the scan by pressing ESC. Answer 'N' to the Question "Return to Command Mode?".

b. A menu will appear in the left hand part of the screen. The first choice in the menu is whether to continue sampling or to alter the controller. This menu works the same as the others in UnkelScope and the options are described in detail below.

c. When you are done making the changes, return to the Continue Sampling mode, press return and answer any questions that appear in the upper left of the screen appropriately to restart a scan or to continue plotting from where you left off. Remember that while you are in the sample and display mode you are 'erasing' old data when you take new data.

3. Details of the Modify Controller Menu

Mode. This allows you to return to sampling or to activate the remainder of the menu.

Command (Source). This will show the source for the command signal. You can not alter the source, but if SetP is selected, you will be able to alter the value of the setpoint.

Val. If shown this is the value of the Set point. You can alter it by backspacing over the value and then retyping the correct number.

Feedback (Source). For controllers other than Open Loop, this will show the source for the feedback signal. It is not possible to alter the source of the feedback signal.

Output (Channel). This will show where (which channel) the output signal is going to. You can not alter this selection.

Gain, or Prop, Integ, Deriv. These are the controller parameters and can be modified by erasing them and typing in the new values.

Reset. For the PID controller, you can chose to 'reset' the controller before continuing.

```
(Abort by Keyboard) Return To Command(Y OR N):n          UnkelScope (c) MIT 1984
   Trace 1:          5.00E+00+---------+---------+---------+---------+---------+-
                             !
SCAN AT 27-Aug-1987 11:19:46!
 NO CLOCK FAULTS             !
                             +
Trace 1:                     !
Voltage                      !
Analog  0                    !
                             +
                             !
Mode [Change Controller]     !
Type [Proportional]          !
Command [Tr 1]               +
Feedback [Tr 2]              !
Output [D/A 0]               !
Gain: 1.00E+00               !
                             +
                             !
                             !/
   Trace 1:           .00E+00!
                             +---------+---------+---------+---------+---------+-
{Trace 1:Sampled}            .00E+00        Time (seconds)                5.00E+01
```

Representative Display of the Display Screen while altering the controller values.

13.3 Stepping or Traversing Coordinated Sampling

It is useful to coordinate the sampling process with the stepping or traversing of an experimental parameter. Figure 13.7 shows schematically, how the software and some additional hardware are used to obtain the temperature profile across the output of a blow dryer. In Figure 13.7a the traversing hardware consists of a stepper motor driving the gear of a rack and pinion translation stage. In this case the position is measured by a potentiometer that is turned directly by the stepper motor. In Figure 13.7b, the traversing hardware is the horizontal (x) axis of an x-y plotter. In this case the plotter is assumed to produce the proper motion, i.e., the position is determined directly from the voltage applied to the plotter input.

In each case the sensor the sensor is a thermocouple/thermocouple amplifier combination that produces an output of 10 millivolts per °C. The sections below describe the software interface for stepper motors and for analog driven devices.

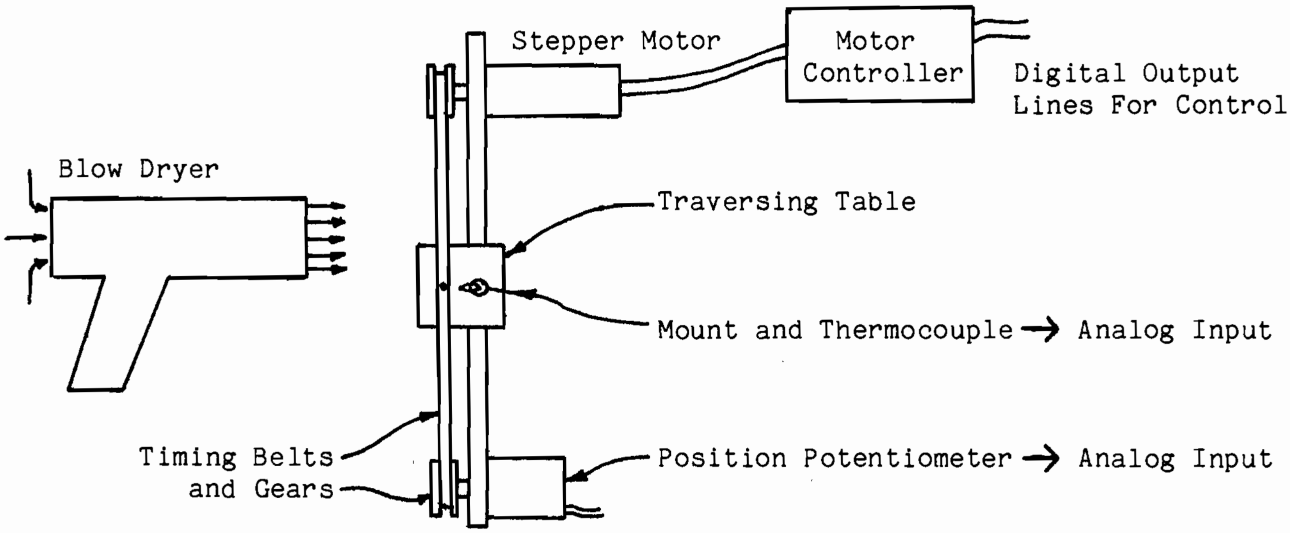


Figure 13.7a Schematic of Traversing Coordinated With Sampling Using a Stepper Motor for Positioning.

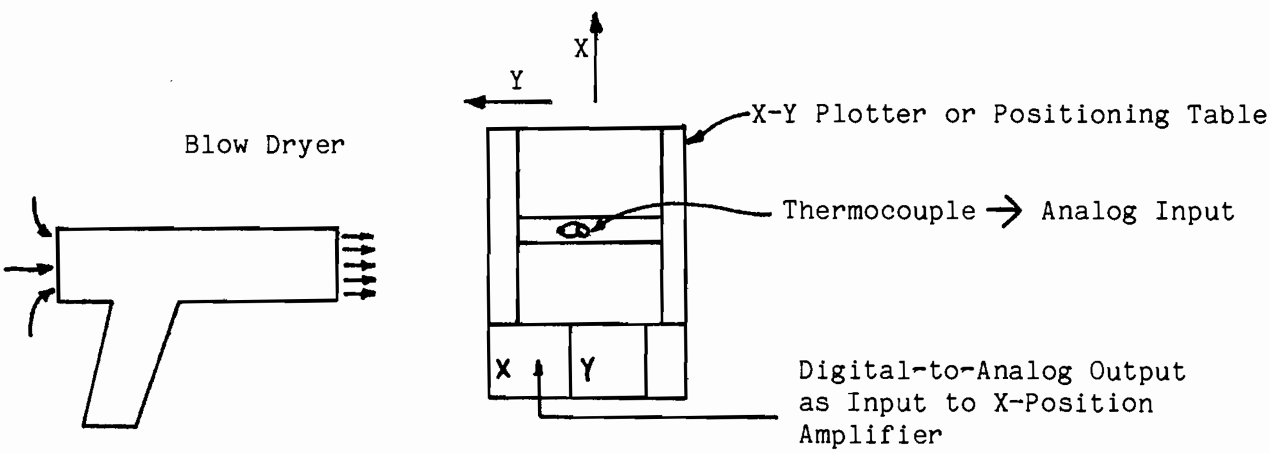


Figure 13.7 Schematic of Traversing Coordinated With Sampling Using an Analog Driven Table.

13.3.1 Setup for Stepper Motors

The software uses two of the digital output lines to control the motion of a stepper motor. The stepper motor inputs must be of TTL logic type to be compatible with the digital output hardware. A signal corresponding to the "position" of the device can be used to limit the rotation of the stepper motor to within defined bounds; this signal is also needed to determine how the change in stepper motor relates to change in experimental conditions. If the "position" signal is provided the device can also be reset to the high or low limit of travel.

The Hardware. The digital output lines are "TTL Logic" and have two states. The low state corresponds to a voltage less than 1.0 volts; the high state corresponds to a voltage of 3.5 volts. The stepper motor moves when a high-to-low transition (or a low-to-high transition) is detected on one of its lines. For a forward/backward motor setup, shown in Figure 13.8a, transitions on one line cause the stepper motor to go in the 'forward' direction while transitions on the other line cause the motor to step in the opposite 'reverse' direction. For a step/direction stepper motor, shown in Figure 13.8b, one line determines the direction of motion (i.e., high makes it go forward, low backwards) while transitions on the second line cause the motion.

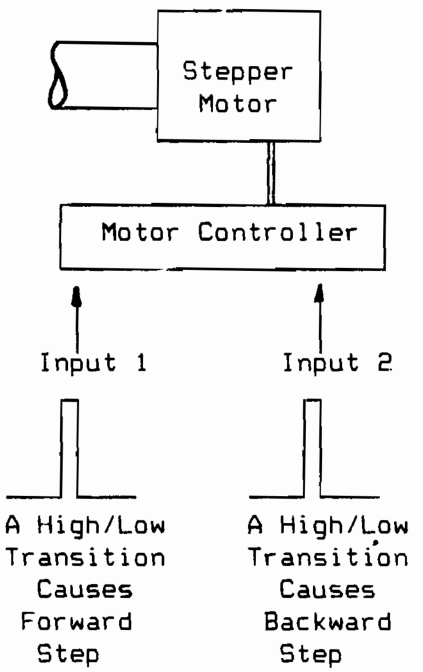


Figure 13.8a Setup for a forward/ backward motor.

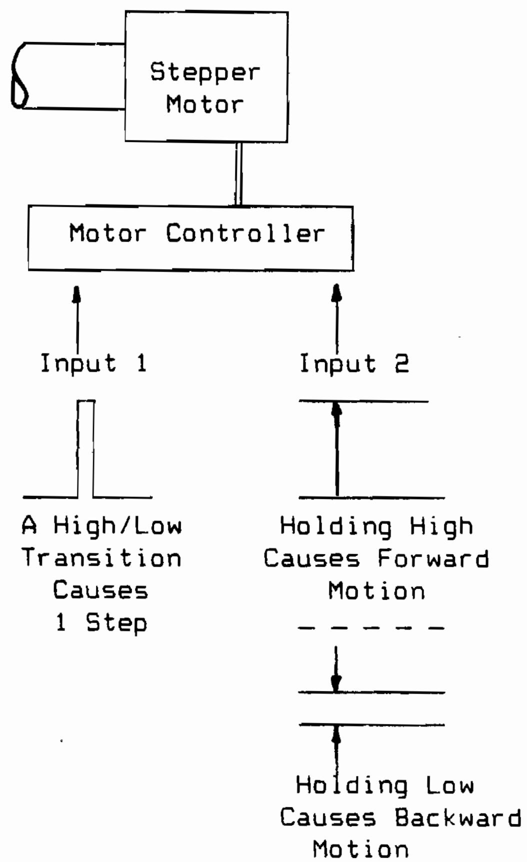


Figure 13.8b Setup for a step/ direction motor.

Note that the link from the computer to the stepper motor is strictly through the TTL lines and the stepper motor control/power circuitry is not provided by the computer.

For the IBM DACA the connections are as follows:

Forward/Backward Type Connections		Step/Direction Type Connections	
Forward	36 / B00	Step	36 / B00
Backward	38 / B01	Direction	38 / B01
Ground	15 / GND	Ground	15 / GND

The Software. The software interface is shown in Figure 13.9a for forward/backward type motors and in Figure 13.9b for step/direction motors.

```
------------------ Processing ------------------
Type [Traversing                  ]

Stepper Type [back/forth] Direction[+]
Steps/group [  4]  Samples/Group [  8]

Sense Input [  0 ] Reset at Start [lo]
Lo Limit[ 2.00E-01]Hi Limit[ 4.50E+00]
```

Figure 13.9a Software interface for forward/backward type motors.

```
------------------ Processing ------------------
Type [Traversing                  ]

Stepper Type [step/drctn] Direction[+]
Steps/group [  2]  Samples/Group [  4]

Sense Input [  2 ] Reset at Start [hi]
Lo Limit[ 1.01E+00]Hi Limit[ 4.00E+00]
```

Figure 13.9b Software interface for step/direction type motors.

This section describes each option given in the software.

Stepper Type. This choses between the two types of stepper motor arrangements as described above.

Direction. This option sets the direction of stepping, [+] for forward stepping and [-] for backward stepping.

Steps/Group. This option sets the number of steps to be taken before data is sampled. The value toggles in factors of 2 from 1 to 128.

Samples/Group. This option defines how many samples of data to take before the next movement of the stepper motor. The value toggles in factors of 2 from 1 to 128.

Sense Channel. This establishes the analog input channel used for determining the "position" of the device being stepped. The software allows choice of any valid analog channel or none. If [none] is specified the software does not check to see if the position is beyond the bounds and the software cannot reset the device to particular limits. Note that an analog channel rather than a particular Vertical Trace is specified. The sense signal does not need to be a sampled channel (although it usually will be).

Reset at Start. This instructs the software what to do at the start of a scan. if [no] is selected the stepper motor is left in whatever position it is in. If [Lo] is selected, the stepper motor is moved in the backward direction until the Lo limit is reached. If [Hi] is selected, the stepper is moved in the forward direction until the hi limit is reached.

In either case the Hi or Lo reset cases, typing ESC will abort the attempt to reach one or the other limit and will start the scan at the current position.

Lo Limit. If a sense signal is selected, the software will insure that the stepper motor will stop being rotated in the backwards (-) direction if the sense signal is less than the value specified here. It is the users' responsibility to insure that the sense signal device is properly configured. If the limit is reached, a message is printed to the message line. The Lo Limit toggles using the left/right arrow keys. If [none] is selected for the sense signal, this value is not used.

Hi Limit. If a sense signal is selected, the software will insure that the stepper motor will stop being rotated in the forward (+) direction when the sense signal is greater than the value specified here. It is the users' responsibility to insure that the sense signal device is properly configured. If the limit is reached, a message is printed to the message line. The Hi value is toggled using the left/right arrow keys. If [none] is selected as the sense signal, this value is not used.

In either case the Hi or Lo reset cases, typing ESC will abort the attempt to reach one or the other limit and will start the scan at the current position.

13.3.2 An Example for Stepper Motors

An example of the use of the stepping-coordinated sampling is shown schematically in Figure 13.10. The hardware consists of a track with a slider driven by a timing belt with the gear rotated by a stepper motor. The motor controller hardware is of the step/direction type. The position is measured using a linear voltage displacement transducer (LVDT) which has an output voltage proportional to the position of the slider rod. The LVDT output is used to fix the starting and stopping position of the traverse.

The stepper motor-driven traversing table is used to move a pitot tube across a jet. The pressure at the pitot tube is measured by a pressure transducer.

The setup configuration for a run is shown in Figure 13.11. The system is set to start the traverse at the low limit and move across the jet in the forward direction. To allow for some signal smoothing, 8 samples are taken at each position and all data is saved. The system is set to display the pressure transducer signal as a function of the position signal. One set of results is shown in Figure 13.12.a. The time variation is shown in the replotting of results in Figure 13.12b, where the pressure signal and the position signal are each plotted as a function of time. This second plot shows the step variation of the position.

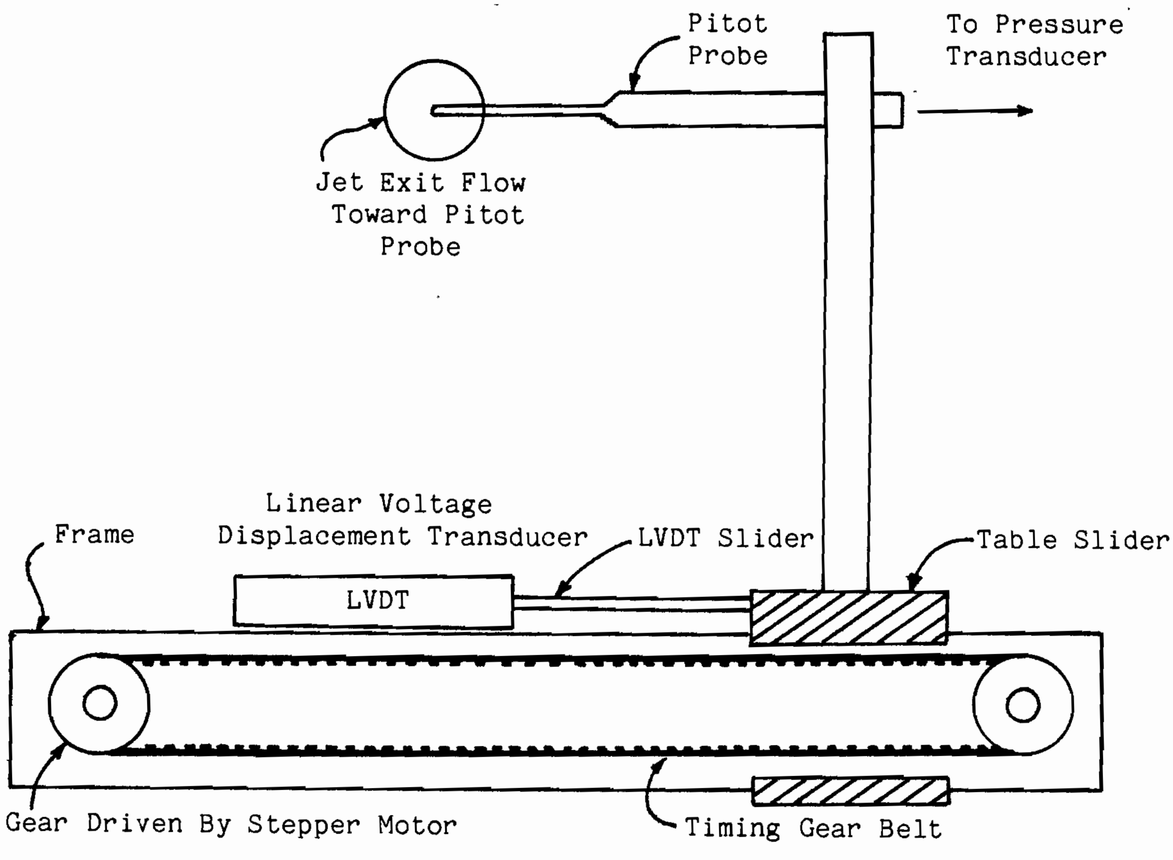


Figure 13.10 Schematic of stepper motor hardware used in the example.

```
——————— Vertical Trace 1 ———————
Source [Analog  1]  A/D Range [+/- 10]
Label  :Pitot Signal
Span   [  2    v full scale]
Range  [ 1.00E+00 to  3.00E+00]

——————— Horizontal Trace ———————
Source [Analog  2]  A/D Range [+/- 10]
Label  :Position Signal
Span   [  2    v full scale]
Range  [ 1.00E+00 to  3.00E+00]

——————— Sampling ———————
Sample Rate [ 0.1 s         10 hz]
[1024] Samples  {Scan Time 1.02E+02 s}
{Real Time Plot} {Processing Active  }

——————— Additional Vertical Traces ———————
Tr Input                          A/D
#  Chan  Label:                   Range
3 [none ]
4 [none ]

——————— Triggering ———————
Mode  [Singl Sweep] Source [Keyboard ]

——————— Processing ———————
Type [Traversing                ]

Stepper Type [step/drctn] Direction[+]
Steps/group [   1]  Samples/Group [   4]

Sense Input [  2 ] Reset at Start [lo]
Lo Limit[ 9.99E-01]Hi Limit[ 3.00E+00]
```

Figure 13.11 Setup configuration for stepper motor example.

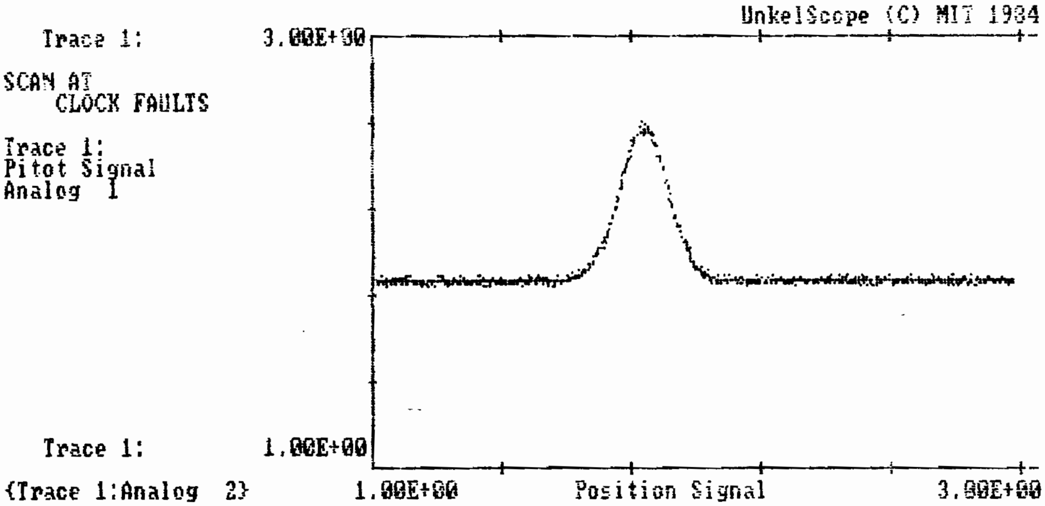


Figure 13.12a Results of traverse as a crossplot.

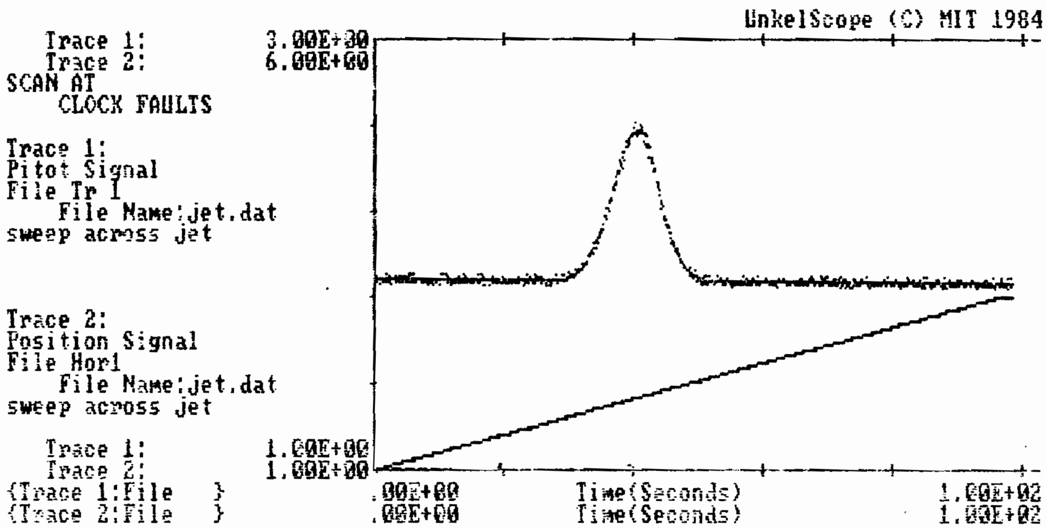


Figure 13.12b Results of traverse as a time plot.

13.3.3 Setup for Analog Driven Systems

The software uses digital-to-analog (D/A) Channel 0 to control the motion of a device whose position is set by an input voltage. If your board has no digital-to-analog channels this option will not appear in the menu. A position servo motor, such as in a stripchart or x-y recorder is one example of such a device. In this case, the software assumes that the device responds precisely to the voltage command, that is, there is no sensing of the actual position.

The Software. The software interface is shown in Figure 13.13.

```
 ______________ Processing ______________
|Type [Traversing                ]       |
|                                        |
|Stepper Type [d/a port 0] Direction[+]  |
| Bits/group [  1]  Samples/Group [  2]  |
|                                        |
|Sense Input [none] Reset at Start [lo]  |
|Lo Limit[ 9.99E-01]Hi Limit[ 3.00E+00]  |
```

Figure 13.13 Software interface for analog driven traversing table.

Stepper Type. Select this as [d/a port 0].

Direction. This option sets the direction of stepping: [+] for forward and [-] for backward.

Bits/Group. This option sets the number of bits to be added to the previous value of the output before data is sampled. The value toggles in factors of 2 from 1 to 128.

Samples/Group. This option defines how many samples of data to take before the next movement of the analog driven device.

Sense Input. The sense feature is not used for the analog driven situation. The program assumes that the hardware responds precisely to the signal.

Reset at Start. This instructs the software what to do at the start of a scan. If [no] is selected the analog output is left at the previous value. If [Lo] is selected, the software starts with the output at the Lo Limit. If [Hi] is selected, the software starts with the output at the Hi Limit.

Lo Limit. The software will insure that the output will not be less than the Lo Limit. If the limit is reached, a message is printed to the message line. The Lo Limit toggles using the left/right arrow keys.

Hi Limit. The software will insure that the output will not exceed the Hi Limit. If the limit is reached, a message is printed to the message line. The Hi Limit is toggled using the left/right arrow keys.

13.3.4 An Example for Analog Driven System

An example of the use of the analog driven table is shown schematically in Figure 13.14. The hardware consists of an x-y plotter, the x direction stage which is used to position a thermocouple in the exit jet of a hair dryer. The x input amplifier is set to a sensitivity of 1 volt = 1 inch, such that the variation of the D/A output voltage by 5 volts gives the desired movement of about 5 inches.

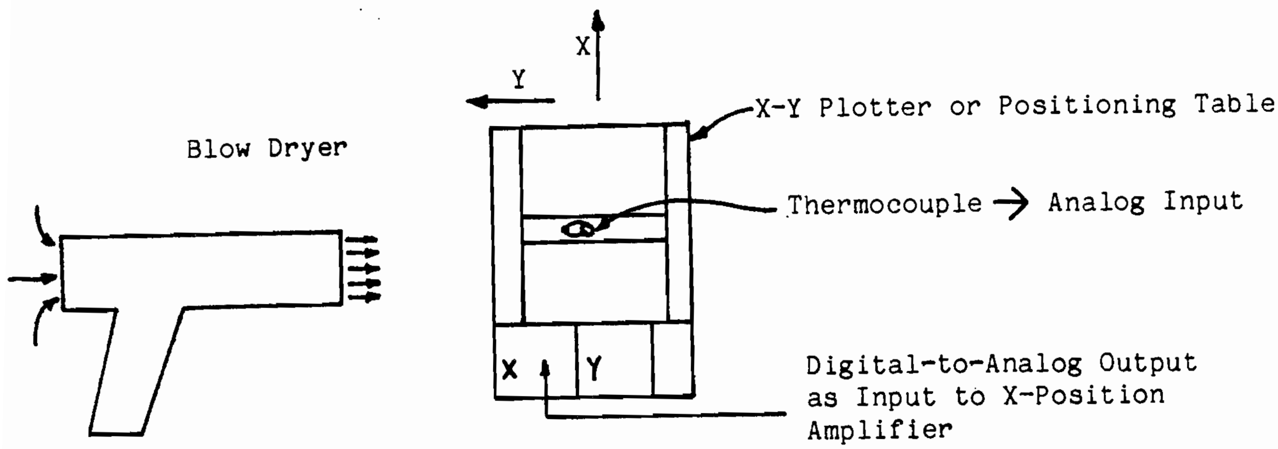


Figure 13.14 Experimental setup for analog driven table example using the traversing software.

The setup configuration for a run is shown in Figure 13.15. The system is set to start the traverse starting at the high limit which corresponds to a position just outside the jet in the backward direction. The device moves a distance corresponding to 4 bits, with 8 samples are taken at each position of the jet. The system is set to display the thermocouple output signal (an amplifier is used to bring the level to 10 millivolts per °C) as a function of the D/A signal, which is assumed to correspond directly to position of the thermocouple. One set of results is shown in Figure 13.16a. The time variation is shown in Figure 13.16b, where both the thermocouple and position signals are plotted as a function of time.

```
——————— Vertical Trace 1 ———————
Source [Analog  0]  A/D Range [± 10  ]
Label  :Temperature Signal
Span   [  1    v full scale]
Range  [  .00E+00 to  1.00E+00]

——————— Horizontal Trace ———————
Source [Analog  2]  A/D Range [+/- 10]
Label  :Position Signal
Span   [  2    v full scale]
Range  [ 1.00E+00 to  3.00E+00]

——————— Sampling ———————
Sample Rate [ 0.1 s         10 hz]
[1024] Samples  (Scan Time 1.02E+02 s)
(Real Time Plot) (Processing Active  )

——————— Triggering ———————
Mode  [Singl Sweep] Source [Keyboard ]

——————— Processing ———————
Type [Traversing              ]

Stepper Type [d/a port 0] Direction[+]
 Bits/group [  1]  Samples/Group [  2]

Sense Input [none] Reset at Start [lo]
Lo Limit[ 9.99E-01]Hi Limit[ 3.00E+00]
```

Figure 13.15 Setup for analog drive table example.

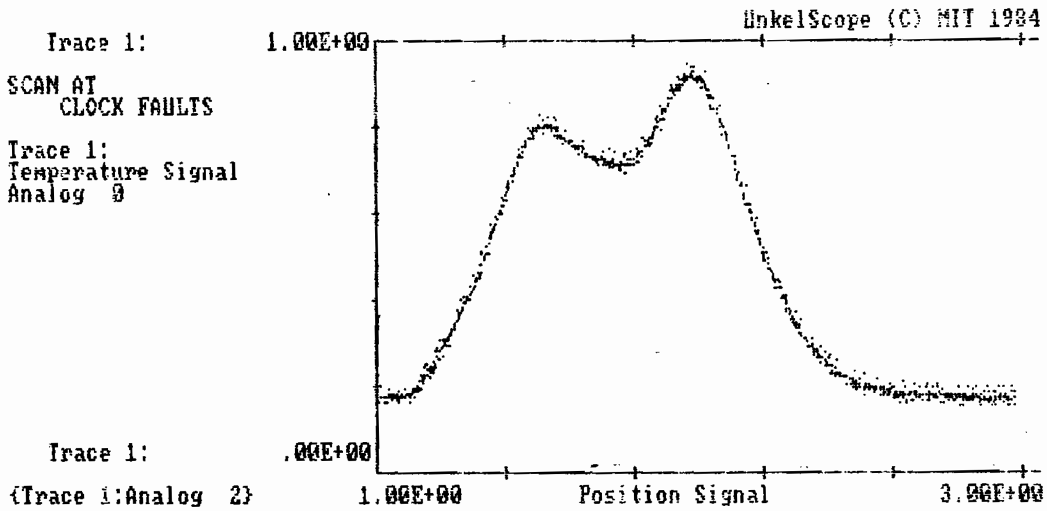


Figure 13.16a Crossplot of traverse across the jet.

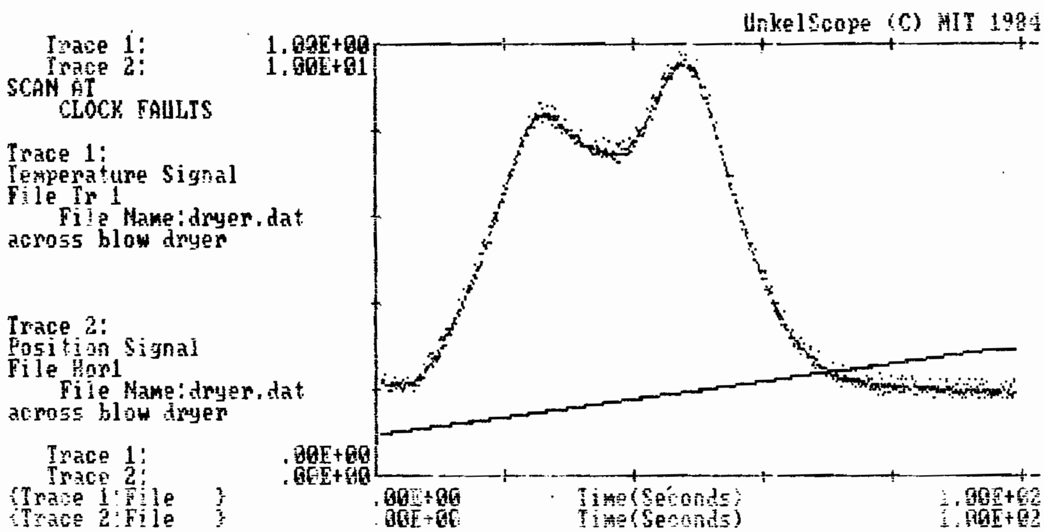


Figure 13.16b Results replotted as a function of time.

14.0 Online Data Reduction

UnkelScope allows for online linear conversion of sampled data. This option is selected as a choice to the User Processing Section of the SETUP MODE. To select the linear conversion option, move to the User processing line of the Setup mode and toggle to the [Linear Conversion] choice. Then move down to specify, for each sampled signal, the slope and intercept for the conversion. The conversion is applied only to newly sampled data, that is the data retrieved from files will not be affected. The default value for slope is 1.0 and the default value for the intercept is 0.0. Figure 14.1 shows the setup screen. Conversion can be made for each of the 4 vertical traces and the variable used as the horizontal trace (time or a sampled signal.)

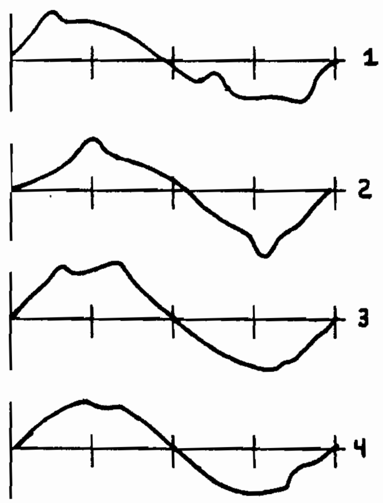
Entering Numbers. The numbers are entered as up to 9 characters in either "i", "f" or "e" FORTRAN type format. For example the number 92 could be entered at 9.2E+1 or 92. or 92 or 920E-1. The return key (or typing the 10th character) signals the end of a number. When a number is being entered, the Delete key deletes the last character and Ctrl/U deletes all the characters describing that number. The program tries to interpret numbers with peculiar characters but may not be able to. If the number is not in the correct form a message is printed to the messsage line.

```
---------- Vertical Trace 1 ----------      ------- Additional Vertical Traces -------
Source [Analog  0]  A/D Range [+/- 10]      Tr Input                              A/D
Label  :Temperature (C)                     #  Chan  Label:                     Range
Span   [100    v full scale]                3 [  1  ]Pressure (psig)        [+/- 10]
Range  [  .00E+00 to  1.00E+02]             4 [none ]

---------- Vertical Trace 2 ----------
Source [Analog  3]  A/D Range [+/- 10]
Label  :Position Signal
Span   [ 10    v full scale]
Range  [-5.00E+00 to  5.00E+00]             -------------- Triggering --------------
                                            Mode  [Singl Sweep] Source [Keyboard ]
---------- Horizontal Trace ----------
Source [Time      ]
Label  :Time (minutes)                      -------------- Processing --------------
Span   [1000 s full scale]                  Type [Linear Conversion       ]
Range  [  .00E+00 to  1.00E+03]             {Only Currently Sampled Data Altered}
                                            Vert 1 slp: 1.00E+02 intcpt:-3.00E+00
-------------- Sampling --------------      Vert 2 slp: 1.00E+00 intcpt:  .00E+00
Sample Rate [ 50  s              ]          Vert 3 slp: 3.22E+00 intcpt: 2.10E+00
[1024] Samples  {Scan Time 5.12E+04 s}      Vert 4 slp: 1.00E+00 intcpt:  .00E+00
{Real Time Plot} {Processing Active  }      Time   slp: 1.67E-02 intcpt:  .00E+00]
```

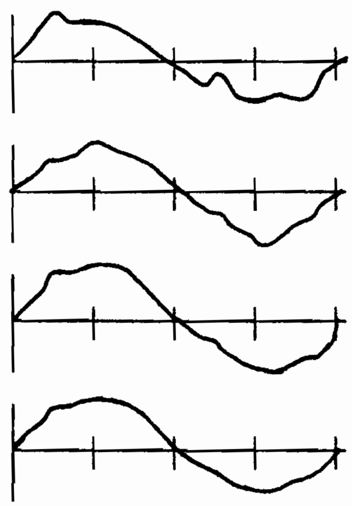
Figure 14.1 Setup configuration showing the online linear conversion options.

14.1 Synchronous Averaging

Synchronous averaging is a technique for smoothing out noise in a signal that can be repetitively sampled, with a trigger signal provided for starting the sampling. For example, suppose that a noisy sinewave is produced by a device and UnkelScope set to trigger at the start of one of the sinewaves. The situation is shown schematically in the sketch below. With the Synchronous Averaging Option, UnkelScope will for each successive scan, average the values at each time with all previously sampled scans. For sampled values of Vertical Traces 1 and 2, the average values will be plotted on the screen. The situation is shown schematically below. For synchronous averaging to work properly, the signal must be repetitive and the trigger must start sampling at the same point on the trace.



Data of Each Scan



Synchronous Average

The synchronous averaging works only for Sampled values on Vertical Traces 1, 2, 3, and 4 and for Horizontal Trace 1. Any data sampled in Vertical Traces 5, 6, 7, and 8, will contain data from the final scan.

15.0 Graphical Editing

Graphical Editing provides several features to allow you to obtain numerical results related to your data while you see the data plotted on the screen. These results can be displayed on the screen, and then printed on your printer and put into a disk file. There are two cursors which can be moved through either of two traces shown on the screen; the two traces can be data just sampled or data retrieved from previously stored data. The activities that can be performed using the Graphical Editing process are:

a) Moving the Cursors while displaying the current cursor values and differences between the values at the two cursors;

b) Calculations based on the current cursor(s) position (eg. enclosed areas, curve fits);

c) Saving values into a disk file and/or printing them;

d) Defining Local Macros to automate your graphical editing procedures (Release 2.25 and later only); and,

e) Defining new data files from existing data (Release 3.00 and Later Only)

These functions are controlled by a combination of UnkelScope type menus, arrow keys, and function keys. The operations are described in detail in the sections below, and are summarized in the Short Guide Sheet 'Graphical Editing.'

15.1 Activating Graphical Editing

The graphical editing feature works on the data currently stored in Vertical Trace 1 and/or Vertical Trace 2. For all UnkelScope products, the Graphical editing can be activated from the Smpl/Disply mode under certain conditions. For UnkelScope Level 2+, Graphical Editing can be activated using the Edit Data choice of the Utilities Menu.

To activate Graphical Editing from the Sampl/Dsply option, you must either be a) sampling data using the Single Sweep triggering option, or b) displaying only data from files (i.e. not sampling any data.) If you are sampling data using the Single Sweep option you can activate Graphical Editing by typing Ctrl/E (hold down Ctrl and press E) when you see the message 'Press A to Scan Again. Press Y to Return to Setup' in the upper left of the screen. You will be able to perform the graphical editing features and then return to the same position. If you are displaying previously sampled data, you can activate Graphical Editing by typing Ctrl/E (hold down Ctrl and press E) when the message 'Press Y to Return to Command Mode' is shown upper left of the screen.

With UnkelScope Level 2+, you can also activate Graphical Editing by first moving to the Utilities choice of the main Menu, then pressing Return. Then select the Edit Data choice of the Utilities Menu and press Return. UnkelScope will show the display screen and plot any data according to the Setup configuration.

After activating Graphical Editing, the left portion of the screen will be replaced with the Graphical Editing Menu and basic Display.

15.2 The Edit Data Menu

As shown in Figure 15.1, the Edit Data Menu consists of an option line with choices toggled in the usual way. The choices are:

Return - Scope — selecting this option and pressing the down arrow key will end the session with the Edit Data function and allow you to return to the main menu of the UnkelScope program.

Setup Cursors — Selecting this option and pressing the down arrow key will allow you to alter the choices for the cursor motion.

Move Cursor — Selecting this option and pressing the down arrow key will allow you to scroll the cursors through the data. **Pressing ESC key will return you from the Move Cursor mode to the Edit Data Menu line.**

In the high resolution graphics mode, it is not possible to use reverse video to highlight the choice you are currently selecting. Instead the choice currently being modified is bracketted by a box symbol, while the choices not being modified are bracketted by the [and] symbols.

The next sections describe the Cursor setup menu and the actual operation of moving the cursors.

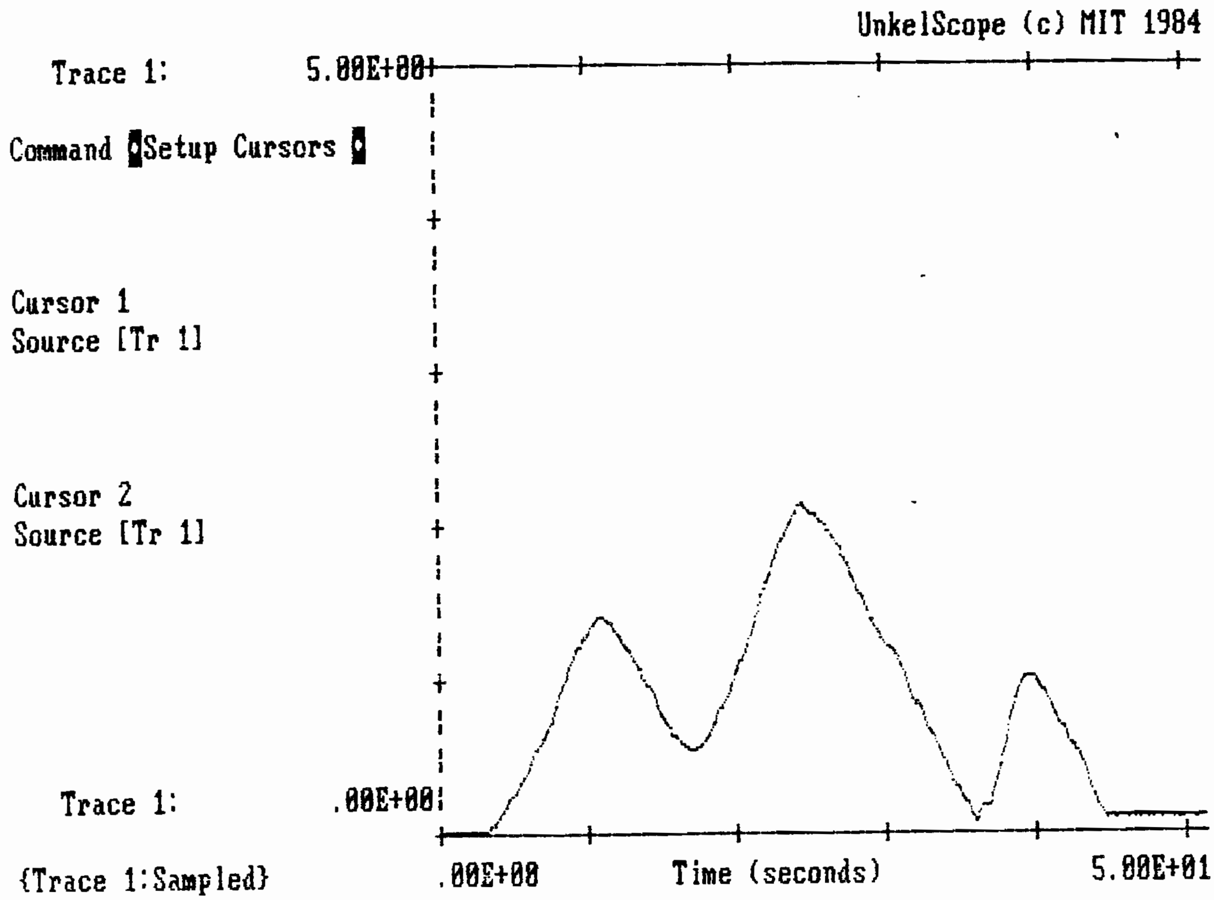


Figure 15.1 Sample of the choices that might be made in the Edit Data Menu.

15.3 Cursor Setup Menu

The cursor menu allows selection of the Cursor Source for each cursor.

Cursor Source. If there is data stored in each of Vertical Trace 1 and Vertical Trace 2, you can select the cursor source to be either of the two traces. If there is data only in Vertical Trace 1, you will not be able to change the choice from the default, which sets each cursor to operate on Vertical Trace 1.

Note: The Cursor Speed option of earlier versions is no longer required as all cursor speeds can be accessed without returning to the Cursor Setup Menu.

When the Cursors are properly specified move back to the Edit Data Main (Command) menu.

15.4 Cursor Motion

After selecting the cursor options as described above enter the cursor motion mode by selecting the option "Move Cursors" and pressing the down arrow key. The cursor - a large crosshair - will be located on the current point, the "Active" message printed next to the appropriate cursor, and the current cursor position values will be displayed. A sample display is shown in Figure 15.2. At this point you can move the cursors, change the active cursor, activate the functions, and return to the Edit Data Command mode. The keys that control the motion of the cursor and their resulting action are shown below.

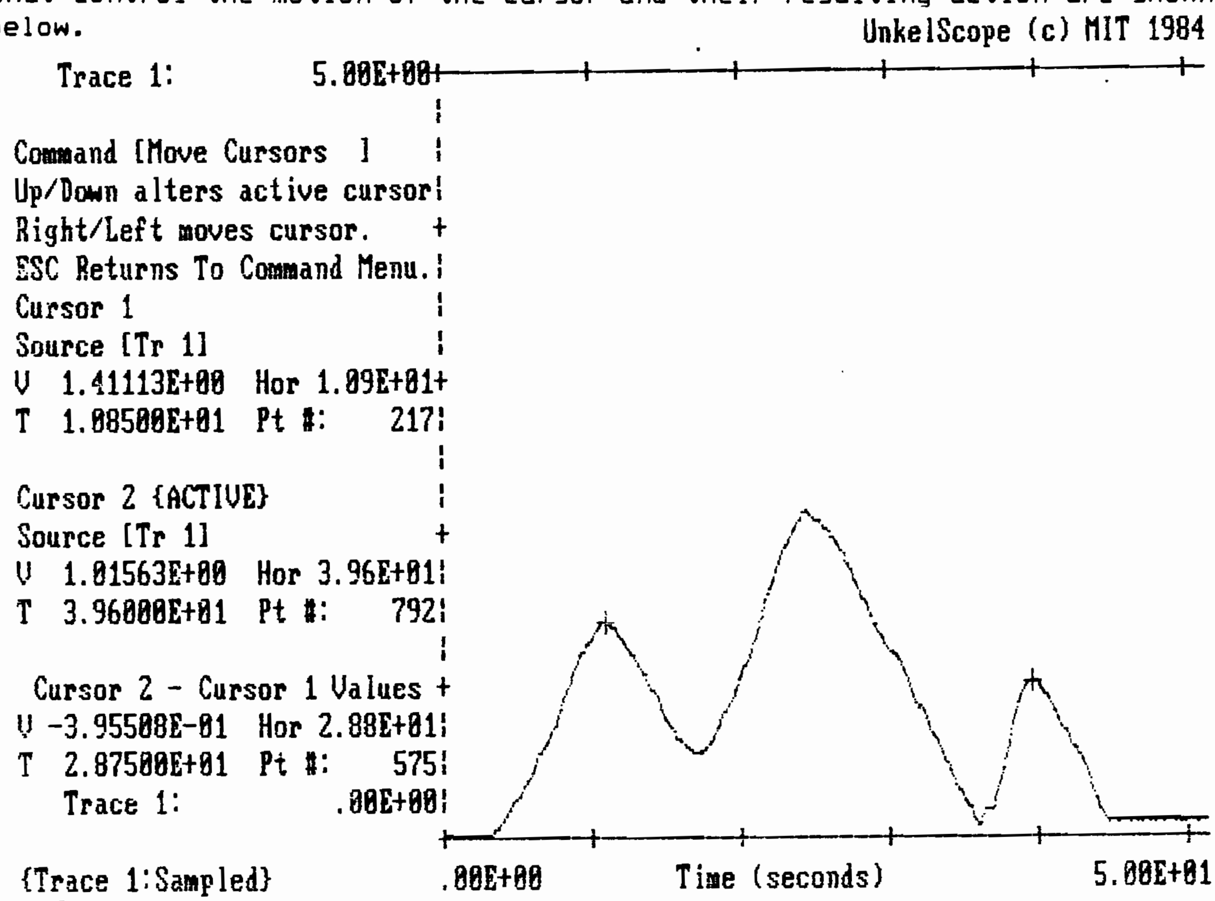


Figure 15.2 Sample trace showing the cursors and the numerical values.

15.4.1 Return from Moving Cursors

ESC - returns to the Edit Data Command Mode.

15.4.2 Change Active Cursor

Up Arrow or Down Arrow	-	switches from Cursor 1 to Cursor 2 or vice versa. The "Active" message is automatically altered.

15.4.3 Move Active Cursor by Point Number

Press Key	While Holding Down none	Ctrl	Shift
left arrow	1 left	4 left	32 left
right arrow	1 right	4 right	32 right
End	last point	-	-
Home	first point	-	-

15.4.4 Move Active Cursor to Next/Previous Peak/Trough

You can also move the cursor to the next/previous peak/trough. The next peak, for example is found by moving the cursor until the Vertical Trace value of point N+1 is less than the Vertical Value of point N. Since the resolution on the plotted screen may be less than the resolution in the values, with a noisy signal, the cursor may stop at positions that do not appear to be peaks.

Move Active Cursor to Next/Previous Peak/Trough

Press Key	While Holding Down none	Ctrl	Shift
page up	-	Next Peak	Previous Peak
page down	-	Next Trough	Previous Trough

15.4.5 Special Cursor Actions

Out of Bounds Values. When a point is out of bounds toward the top or bottom of the screen, it will be plotted at a level just above or just below the grid boundary. The numerical values displayed for that point will be correct even though the actual plotted position is not. When a point is out of bounds to the left or right, it is NOT plotted and no cursor will appear. To warn you of this, the message "out of bounds" will appear next to the Cursor identification in the left hand side of the screen. The numerical values will be correct.

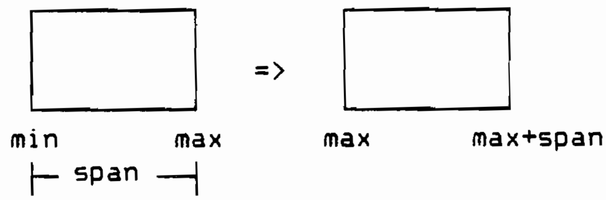
Wraparound. The cursor motion is set to wraparound, that is if you are at the last point in the data set and press the right arrow key, you will move to the first point in the data set.

Returning to the Cursor Command Line. At any time, press ESC to return to the cursor command line. From there you can return to the UnkelScope main command mode.

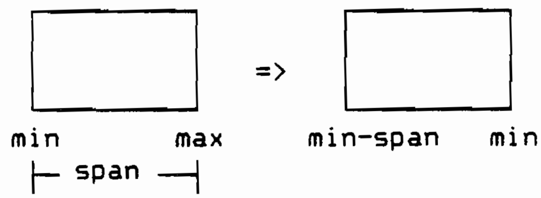
15.5 Changing and Printing the Graphical Display

The Graphical Editing feature allows you to expand and contract and shift the horizontal axis of the Graphical Display. These features are activated using the Function Keys in conjunction with the Shift, Ctrl and Alt keys as described below.

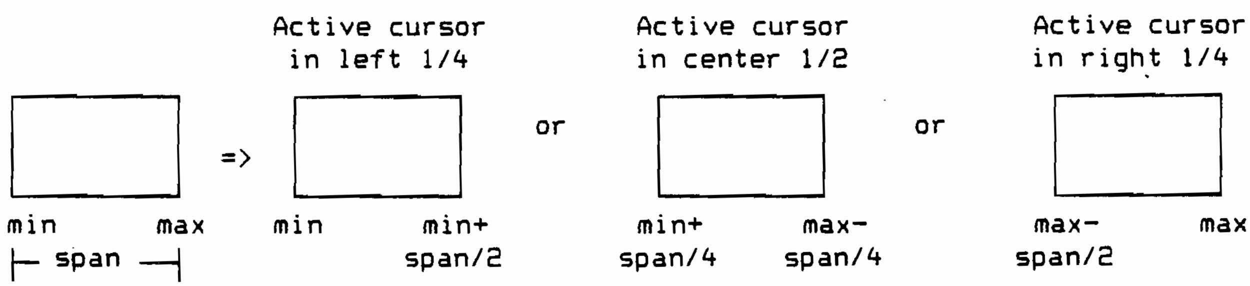
F9 - Move to Next Horizontal Panel. Make the new minimum value equal to the current maximum and the new maximum equal to the old maximum plus the span.



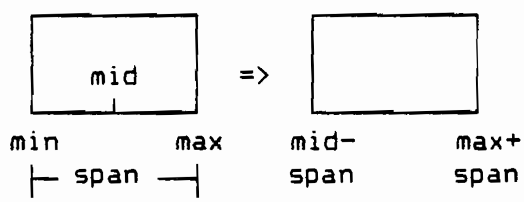
F10 - Move to the Previous Horizontal Panel.



F9 with Shift - Expand the horizontal axis by a factor of 2 about a region that depends on the position of the current cursor.



F10 with Shift - Contract the horizontal axis by a factor of 2 about the current center.



F9 with Ctrl - Return the Horizontal axis to the configuration defined by the Setup Mode, i.e. return to the default plot settings.

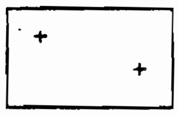
The current screen including both cursors can be printed to your dot matrix printer at any time when you are in the Move Cursor mode by typing Ctrl/F (hold down control and press F.)

15.6 Saving Data In Files <u>(Release 3.00 and Later)</u>

You can save the data shown between the cursors into a file directly from the Graphical Editing Screen. Data from All Valid Traces from Vertical Trace 1 to Vertical Trace 4 will be saved in this file. The points saved will be defined by the cursor positions and the 'padding' instruction chosen. The keystrokes to save data files are shown below.

F1 - Save the data between the cursors (including the points under the current cursor positions) into an UnkelScope data file.

cursor 1 at 257
cursor 2 at 700



The output file will have all the points from 257 to 700 including points 257 and 700.

15.7 Calculated Values and Labels

Graphical Editing can be used to compute values based on the cursor positions, with the values shown in the upper left of the screen. For this purpose the upper left is divided into 4 regions and specific functions will put values into the specific positions defined in the description. Following instructions of Section 15.8, the contents of the Message line can be printed or stored in a disk file for later examination.

15.7.1 Integrate Enclosed Area

You can integrate the area enclosed by the data trace and a line connecting the two cursors as indicated in Figure 15.3. The Area will be computed for the Vertical Trace on the Active Cursor. If the second cursor is on a different trace, then the area will be computed to from the first cursor to the data point number of the second cursor.

The Area value appears following "A =" in the 3rd location of the Message Screen.

F5 - Integrate Area

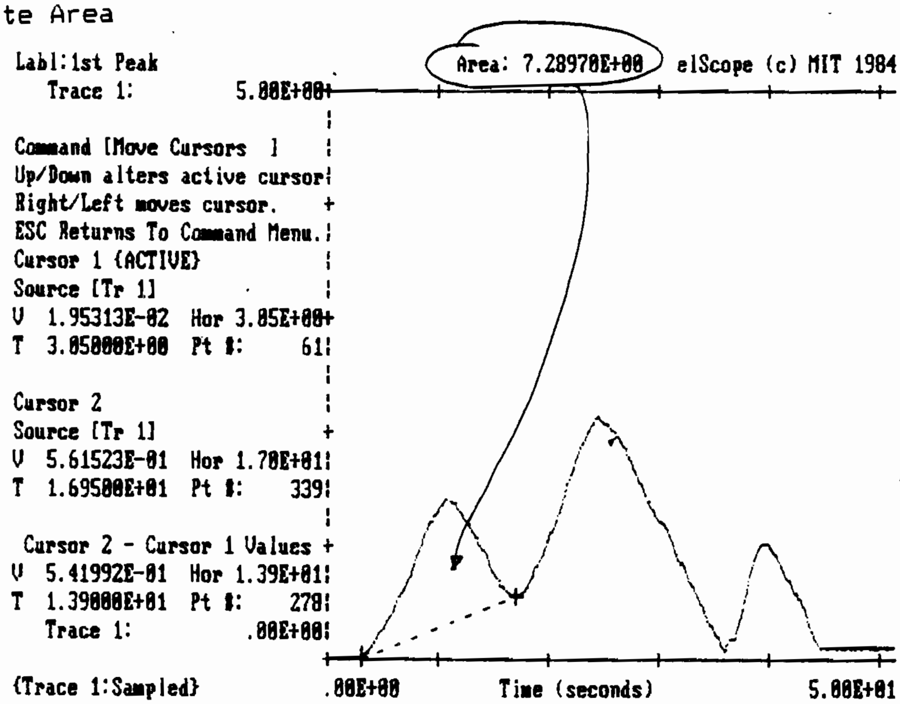


Figure 15.3 Example of Area Calculation.

15.7.2 Current Cursor Values to Message Line

The Vertical, Horizontal and Time Values of the active cursor can be put to the Message line. These values will each appear in the 2nd location of the message line. The specific commands are:

F8 - The vertical value of the active cursor.

F8 with shift - The Horizontal value of the active cursor.

F8 with Ctrl - The Time value of the active cursor.

15.7.3 Linear Curve Fit

You can obtain a linear curvefit to the values between the cursors with the slope and intercept displayed in the 2nd and 3rd locations of the Message line. The Trace of the active cursor will be curvefit, and if the second cursor is on a different trace the curvefit will be between the active cursor and the point corresponding to the other cursor. The specific command is

F6 - Curvefit data between cursors and put the slope and intercept on the message line.

15.7.4 Minimum, Maximum, Average and Rms Fluctuation

The Minimum and Maximum values between the cursors, and the Average and Rms fluctuation of the points between the cursors can be displayed in positions 2 and 3 of the message line. The values will be shown for the trace of the active cursor and if the second cursor is on a different trace, the curvefit will be between the active cursor and the point corresponding to the other cursor value. The specific commands are:

F5 with Shift - Show the Minimum and Maximum values between the cursors in positions 2 and 3 respectively.

F5 with Ctrl - Show the Average value and the Rms fluctuation in positions 2 and 3 of the message line respectively.

15.7.5 Label

A label can be entered into Position 1 of the message line by pressing F7 and then typing the label after the "Lab:" in the left of the message line.

15.7.6 Clearing the Message

The contents of the message line can be erased by pressing F4 with the Shift key down.

15.8 Printing the Message Line

The contents of the message line can be printed to your printer by pressing F3 with the Ctrl key held down. Make sure that your printer is connected and is on line.

15.9 Storing the Message Line into a Disk File

The contents of the message line can be stored in a disk file for later examination. This file will be a 'straight ASCII' file that can be printed or can be read by a program of your design. It cannot be read back directly into UnkelScope. To store the information you must first Open a file. Then you can store the contents of the message line into this file; you can store a sequence of the message lines into the file, limited only by the space available to the disk. Note that the file stays open until you close it, even if you leave the Graphical Editing feature.

The specific commands available to you are:

F3 - Open a file for output. If a file is already opened, you will have to close it before you can open a new one; a message is printed to the message line in this case.

F4 - Close the output file. If no file is open, no action results.

F3 with shift - Put the current message line to the output file.

15.10 Defining Local Macros for Use in Graphical Editing

UnkelScope (Release 2.25 or later) allows you to automate the graphical editing process by defining and re-executing EditMacros. These EditMacros are different than the Procedure capability in that they can be executed only from while Graphical Editing is active. The EditMacros can be stored into a file and retrieved at a later time. The EditMacros are accessed using the Function keys while the ALT key is held down, with the specific functions defined in the description below. Up to 6 EditMacros can be stored each containing up to 40 characters. Note that you cannot call an EditMacro from within another EditMacro or from within a procedure. The EditMacros remain available until they are overwritten from an EditMacro file or you exit the program.

F1 with ALT - Save the current EditMacros into a file. You will be prompted for the file name. Be careful to give a proper filename.

F2 with ALT - Retrieve the EditMacros from an existing file with EditMacros. You will be prompted for the filename.

F3 with ALT - Start Storing an EditMacro. The next character you type must be the EditMacro key your macro will be stored into, and when you have typed in the keys, you must terminate the macro by typing ALT/F4.

F4 with ALT - Finish the EditMacro currently being stored. You must type ALT/F4 to indicate that the Macro is complete. Watch for any error messages in the message line.

F5 with ALT through F10 with ALT - Executes the currently stored EditMacro associated with the key typed. If no EditMacro has been stored, no action will result.

16. Procedures

The Make Procedure and Get Procedure utilities allow you to save the keystrokes associated with a process and re-execute them without retyping the keystrokes. You can also make a procedure that can be executed when you start UnkelScope. A complex procedure can be created out of smaller procedures and a precedure can be executed several times. Data files can be stored as part of a procedure, and the data file names can be altered to store a sequence of files as a test procedure is executed.

Making the procedure consists of a) telling the program to start "remembering" the keystrokes, b) telling the program that you are finished and c) giving it a name under which to store the procedure. When using these facilities, you must observe the cautions described in the sections below. In addition, the sections below describe how to sequence the names of files from one "run" of the procedure to the next.

16.1 Make Procedure

To start the program remembering the keystrokes, move to the UTILITIES choice of the Command line and press the down arrow key. Then move to the Make Procedure choice of the Utility Command mode and press the down arrow key. A message will be printed on the screen, and UnkelScope will start to remember keystrokes. When you are done with the procedure you MUST return to the Make Procedure utility to store the keystroke file.

BEFORE YOU START MAKING A PROCEDURE READ THE MATERIAL BELOW.

<u>Use of Setup Mode.</u> If you are going to sample data or need specific settings in the setup mode, you should store a setup file for each of the cases. When you are making a procedure, you should not move into the setup mode to alter options (see exception below.) Changing options in the setup mode will work only if the setup mode is in exactly the same configuration each time the procedure is called; otherwise the results will be unpredictable. These setup files must be made and stored before you start to make a procedure.

<u>Use single sweep triggering if sampling.</u> If you are sampling data, you should use the Single Sweep triggering mode. Other modes may put you in a situation where you cannot stop the sampling process.

<u>Keyboard Triggered Sampling.</u> When sampling with a Keyboard Trigger the keystroke to initiate the scan DOES NOT come from the procedure file. Thus, you can use the Keyboard trigger to start the sampling when you want, even though the rest of the keystrokes are taken from the previously stored file.

<u>Files.</u> You must make sure that files you expect to exist when making the procedure do exist. While you are making the procedure, you must also make sure that files that you are going to write don't already exist. UnkelScope allows you to automatically overwrite existing files when you execute the procedure, but not when you are making the procedure.

Sequencing of Files. The Get Procedure program will allow you to alter each instance of the character "_" (underscore) that occurs in the procedure file. If you select to repeat execution of the procedure, UnkelScope will alter the "_" in sequence starting from a character you specify.

In this way you can create a "set" of files for each run of the procedure. For example if you use the names:

raw.00_   low.00_   log.00_

as names of stored files while making the procedure, you can later make a run with the sequence of file names altered to

raw.001   low.001   log.001

and in another run of the procedure you could save a sequence with names

raw.002   low.002   log.002

Temporary Files. Many times you will need to make intermediate files that you do not wish to look at and therefore would rather not use up disk space for. To do this you can give the file the name "tempdata." When using the utilities, you would specify the input file name as tempdata. If you want the output file to also be a temporary file, you should enter a blank file name for the output file. (In the utilities, you cannot have the input and output file names the same, but using this technique will get around that restriction.) If you wish to avoid even the writing of the file AND you will not leave the Utilities until you have completed all steps and stored the last file, you can use the file name = which will cause UnkelScope not to store on disk the results of a calculation by one of the Utilities. This saves disk space and also the time to write the files.

Creating a Pause in the Procedure. Sometimes you wish to leave a data display on the screen until you have seen it. To do this you will have to insert an special character into the procedure file. For example, at the end of a single sweep of data, the computer requests that you type an "y" to return to the command mode. If you wish to freeze the screen when you run the procedure, type a "~" (tilda) and then the required "y". When the procedure is run, it will pause at this position, print the message "Press Spacebar To Continue" in the message line and then wait until you press a character before continuing with the stored procedure.

Retrieving Files as Input to the Setup Mode. If you wish to retrieve the files of a particular run when you are replacing the _ character, you must type these names while making the procedure, rather than storing the file names within a setup file. To do this, store the proper setup file, complete with the proper scales etc. but with a blank for the file name. Then while making the procedure, get the setup file and move down to modify the file name to the appropriate name. Then move back to the command mode and proceed normally.

Limit on size of the Procedure. The number of keystrokes that can be stored is at least 140, but may be higher. In the case your procedure was too long, a message will be printed when you try to store the procedure as a file.

Do not use Get Procedure or Edit Data utilities. You cannot call up an existing procedure while you are making a procedure. You should not use the Edit Data Utility while making a procedure.

Closing the Procedure

Do Not forget to close and store the procedure when you are done. When you close the procedure you will be prompted for the following information.

File Name: Type the desired file name.

Special Beginning. When you close the procedure you will be able to select one of three special beginnings to be used each time you execute the procedure:

a) Nothing - the procedure will start executing as soon as you enter all the information about it.

b) Wait For Space Bar - after entering the last information about the procedure, UnkelScope will pause until you press the spacebar (actually any character.) This may be useful to coordinate activities when executing a procedure repetitively.

c) Start Up - this prepares the procedure to be executed as the very first thing UnkelScope does. If you chose this option you cannot use select the file from the Get Procedure Utility. To use a procedure file set-up as a start up file you should invoke UnkelScope specifying the file name after the Scope command. For example, if you name the file "starts.pro" you would start UnkelScope with the command:

```
C> scope starts.pro
```

16.2 Get Procedure

To run a previously stored procedure move to the Utilities choice of the command menu and press the down arrow key. Then move to the Get Procedure choice of the utility command menu and press the down arrow key. The screen will blank and you will be asked for the procedure file name and other information. A sample screen of using the get procedure is shown in Figure 16.1.

When the procedure is modified, the program will print a message in the message line telling you to press the spacebar when you wish the procedure to begin.

When the procedure is completed, you will be left in the Utility Command line mode with the current choice of Return-Scope.

Inputs for Get Procedure

Number of Procedures in Sequence. Toggle this option to chose the number of procedures you wish to "stack" together. You can stack at least 4 procedures, but the total number of keystrokes must be less that the maximum allowed for the program (at least 140 and likely more.)

Input File Names. Type the names of the procedure files that you wish to use. If a file you request does not exist, you will be given the opportunity to examine the current directory. You will be informed if the procedure is too long.
Times to repeat sequence. Toggle to select the number of times to repeat the sequence of procedures. Specifically you can repeat the procedure 1,2,3,4,5,6,7 or 8 times with the ability to automatically sequence file names (see below.) Or you can repeat the sequence 128 times or 16,384 times without the ability to automatically sequence the file names.

Alter _ to. This option allows you to change each instance of _ in the procedure file to another character, or if you select the auto increment option to a set of characters starting with this character. You must insure that the character(s) is a valid character for a file name. Numbers and characters will always work, but some other characters will not. Remember that DOS does not distinguish between upper and lower case when dealing with file names.

Auto Increment File Name. If you select to repeat the sequence of procedures you may wish to alter the file name for each successive case. UnkelScope will start with the character above and sequence to new characters in order. If you chose "1" as the character above you will get the sequence 1,2,3....9. If you start with an "a" (or "A") you will get the letters in sequence. It is your responsibility to insure that all characters selected are valid characters for a file name.

Execute in Quiet Mode. Normally as the procedure file is executed you see the screen changes as if you were typing the characters through the keyboard. You may chose not to see the intermediate information on the screen; this will make the procedure execute slightly more quickly. If you select the "quiet" mode, the data plots will still be shown, but most of the text screen manipulation will not be displayed. Depending on what you do with the procedure, the screen may be blank for extended periods of time. When the procedure is completed, a message will be printed to the upper left of the screen to notify you that control is back to the keyboard.

Automatic Over Write of Existing Files. It is sometimes useful to give the procedure the ability to automatically write over files that are already on disk. The user must be careful that a necessary file is not inadvertently destroyed. If you do not give the procedure permission to over write files you must insure that files written by the procedure do not exist at the time the procedure is executed. If a file exists that should not exist (and no over write priviledge is given) the procedure may not run properly.

Aborting a Procedure

You can abort a procedure by typing a Ctrl/P (hold down the Ctrl key and press P). The next time UnkelScope looks for a character, it will display a prompt in the upper left corner to give you the opportunity to stop the procedure. Note that the prompt will not appear until the program tries to get the next character, so you will not abort an operation already started. For example, if you type a Ctrl/P just after UnkelScope starts to make a Power Spectral Density calculation, the calculation will be completed before you are given the chance to abort the procedure.

```
                                              UnkelScope (c) MIT 1984
 Get  Proced



      Number of Procedures in Sequence [ 2]
                   Input File #1 name:part1.pro
                   Input File #2 name:part2.pro




           Times to repeat sequence [  4]
                    Alter _ to :1        (Previous Character was :_)
           Auto Increment File Name [yes]
              Execute in Quiet Mode [ no]
Automatic Over Write of Existing Files [yes]
```

Figure 16.1 Sample screen showing the use of the
Get Procedure utility.

17.0 Facts

The facts utility provides you with some simple facts about a specified data file. Figure 17.1 shows a typical screen with the results provided by the facts utility. Facts shows the file label and the minimum and maximum value for each valid trace. In addition, facts computes a linear least squared curvefit to the data. For each valid trace the fitting equation is given for the Vertical Trace vs. the Horizontal Trace and for the Vertical Trace vs. Time. If the data was not taken as a cross-plot the values in the Horizontal trace will be a repeat of the time values. Note that the curvefit is done for the entire data set.

```
                                                    UnkelScope (c) MIT 1984
Facts About
      Input file name:junk.dat
Label from Input File:varying signal       1-Jan-1980  0: 2:30




Linear CurveFits: Vert vs Horizontal    Linear CurveFits: Vert vs Time
Vert1=+1.326E-02 * horz1+2.502E+00      Vert1=+1.326E-02 * time1+2.502E+00



 Trace   Minimum   Maximum   Average    Trace   Minimum   Maximum   Average
Vert 1   9.77E-03  5.14E+00  2.84E+00   Horz 1  5.00E-02  5.12E+01  2.56E+01
```

Figure 17.1 Sample of the information given by the Facts Utility.

A

B

C

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X
Y
Z

UnkelScope **Level 1** — **Reference Sheet**

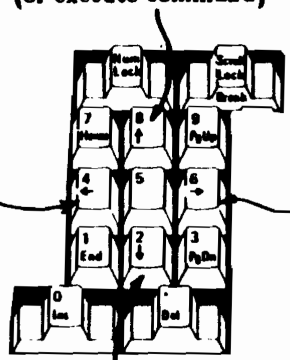
To Run UnkelScope Type:

SCOPE

To Position Cursor or Make Selections: Use Arrow Keys of Special Keypad[*]

move up
(or execute command)

Toggle previous (Higher)



Toggle next (Lower)

move down
(or execute command)

Other Special Keystrokes

Home	- Returns to Command Line When in Setup Mode.
Ctrl/←[**]	- Equivalent to 4 ←.
Ctrl/→	- Equivalent to 4 →.
PgUp	- Equivalent to 4 ↑.
PgDn	- Equivalent to 4 ↓.
Ctrl/g	- Abort operation.
Del	- Delete 1 Character of Typed Input.
Ctrl/u	- Delete Entire String of Typed Input.

[*]Pressing the *Num Lock* key toggles the special keypad between using numbers and using the arrow keys.
[**]Ctrl/*character* means hold down the *Ctrl* key and press *character*.

Summary of Commands

Quit Quit program; lose unstored data.

Smpl/Dsply The sample and display mode takes and displays data according to specifications of the **Setup Mode**.

Setup Mode In the setup mode you specify how, what, where, and when data is taken.

Save Setup *Save Setup* stores the sampling specifications in a disk file for later use[1].

Get Setup *Get Setup* retrieves the specifications stored earlier with *Save Setup*.

Save Data Saves[2] all sampled data on disk[1]. A 20 Character label can be entered to aid record keeping.

Prnt Setup Prints the Setup Screen.

Prnt Dsply Makes a hardcopy of the Display Screen.

Analog Input Connections

J4 Connector Pin Locations
(Top of Computer)

Signal Name	Pin #		Signal Name
D/A Out 1	1	2	D/A Out 0
		4	Analog Ground
A/D In 0 Low -	5	6	A/D In 0 High +
A/D In 1 Low -	7	8	A/D In 1 High +
A/D In 2 Low -	9	10	A/D In 2 High +
A/D In 3 Low -	11	12	A/D In 3 High +
Dig. Ground	15		
Ext Trig Bit	35	36	Dig. Out Bit 0
	37	38	Dig. Out Bit 1
	39	40	Dig. Out Bit 2
	43	44	Dig. Out Bit 3

[1]A legal filename has two parts FILENAME and EXT (Extension), where FILENAME is 8 characters or less and EXT 3 characters or less. The whole name is typed using a '.' to separate the two parts. UnkelScope allows *only 10 total characters* including the '.' Use only letters(lowercase converts to uppercase) and numbers; do not put blanks in the middle of the name. Example Filename: 'NAME00.DAT'.

[2]Data saved can be retrieved in *Setup* at *Vertical Traces 1 or 2*.

Setup Mode

Vertical Trace 1, 2, 3, and 4

Source. Choose to sample new data from any Analog Signal Channel or retrieve data stored earlier from *Save Data* command.

Label. Type an indentifing label up to 20 characters long. Use Delete and Ctrl/U to Correct Mistakes.

Span and Range. Select the upper and lower bounds of the plot. *Range* positions the upper and lower bounds such that they will differ by *span*.

Span. Select the maximum minus the minimum value displayed on the plot.

Range. Select the limits of the span. Select the lower limit, and the upper limit will be higher by the *span*.

Horizontal Trace

Source. Choose Time or an Analog Signal as the horizontal scale. Select *Label*, *Span*, and *Range* as in Vertical Traces.

Time Controls the duration of a scan and how fast samples are taken.

*Time Between Samples.*Select the rate of sampling.

*Number of Samples.*Select the number of samples.

The duration of the scan is the product of the *Number of Samples* and *Time between Samples*.

Triggering

Controls when the sampling begins. Select triggering to synchronize the sampling with events in your experiment.

Mode. Select from *Auto* where the sampling begins immediately, and *Single Sweep* where the scope waits for a synchronizing signal.

Source. Syncronize with the keyboard or with an analog signal with specified *Slope* and *Level*.

Level. Scope will wait until the Analog signal reaches the specified voltage to begin sampling.

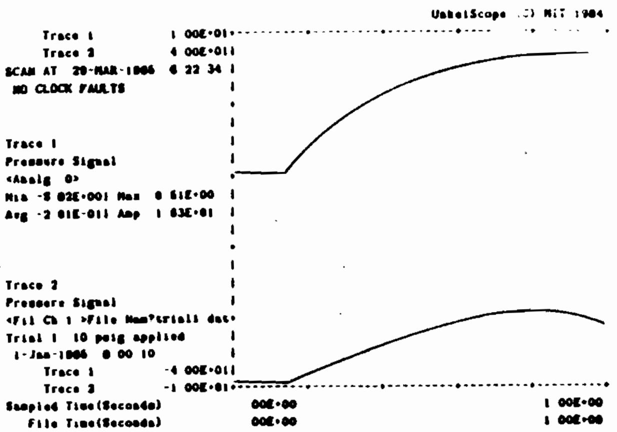
Slope Use + and - to trigger as the signal rises or falls respectively.

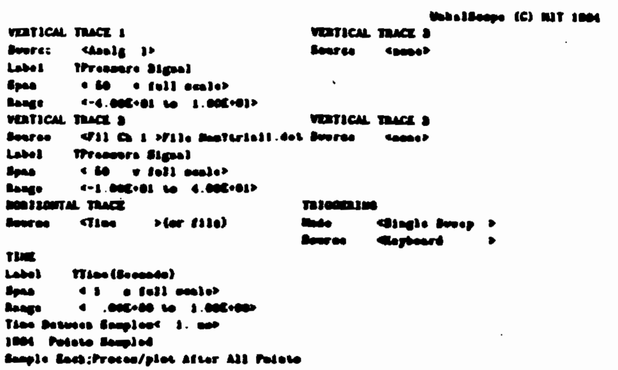
Sampling Strategies

Sample/Proces/Plot Each Point. Data at each time will be sampled and the value plotted in real-time.

Sample Each;Proces/plot After All Points. One full sweep of data values will be taken and then the data will be plotted.

Example: Compare Sampled Data to Previously Stored Data





Storing and Retrieving Data Files

Data files store the data and labels for each trace. The the time, date and a 20 character label are also saved.

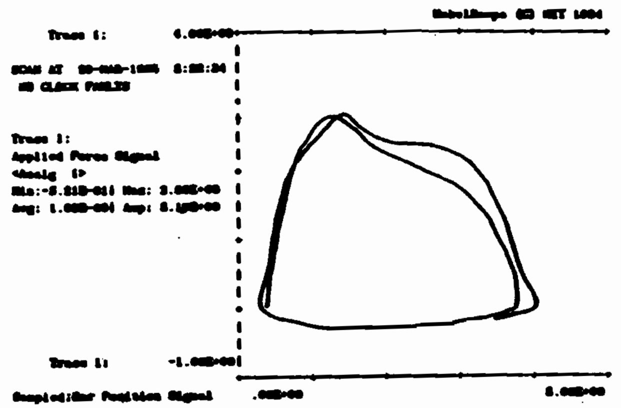
Save Data

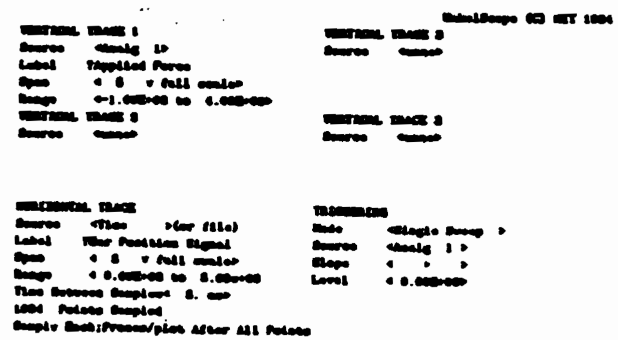
In the command mode, move to the *Save Data* command and press the down arrow key. Type the file name (See footnote[1] other side) and then type the 20 character label. All traces with data will be saved in the file.

Retrieve Data

Previously stored data files are retrieved in Vertical Traces 1, or 2 or in the Horizontal trace. To select the trace you want from a file, move to the *Source* line and toggle to one of the selections *<Fil Tr #>* where # is the number of the trace you want. You can also select *<Hor Tr #>* to select the Horizontal trace.

Example: Cross-Plotting One Voltage Against Another Voltage





File Errors and Directories

If you specify an invalid filename, request a file that doesn't exist or try to use the wrong file type (use a SETUP file where a DATA file is needed), an error message will be printed, and you will be allowed to see a directory. Then you will be allowed to enter a new file name.

Off Line Processing

To process data with your own routines, save the data on disk using *Save Data*. See the UnkelScope Guide for information about using Fortran, Basic, Lotus 123^{tm}, and RS/1^{tm}.

Service and Help

For Software Assistance and Hardware Advice call:

Unkel Software, Inc.*
62 Bridge Street
Lexington, MA 02173
(617) 861-0181

Functi Applied to Data Files
(Minimum, Maximum, Average Values also Reported)

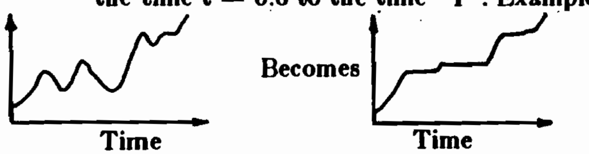
Integral vs. Time. Uses the trapezoidal rule.

Derivative vs. Time. Uses centered 3 point formula except at end points, where an off-centered 2 point formula is used.

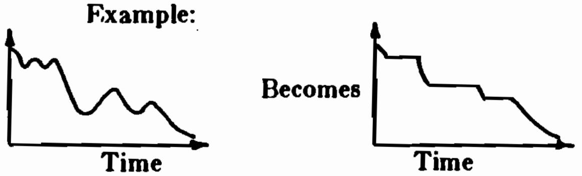
Subtract DC. Finds the average value of the signal and subtractss it from all points.

Absolute Value. Takes the absolute value of each value.

Peaks From Start. Sets the data value for each time "T" equal to the maximum value from the time t = 0.0 to the time "T". Example:



Peaks From End. Sets the data value for each time "T" equal to the maximum value from the final time backwards to the time "T". Fxample:



Floor(minimum). Specify minimum. Sets each data value less than the minimum to the minimum.

Ceiling(maximum). Specify maximum. Sets each data value greater than the maximum to the maximum.

Natural Log. Takes the natural log of each data value. For negative input data the output value is set to -1.e22.

Exponential. Takes the exponential of each data value. For input data values greater than 50., the output value is set to 1.e22. For input data values less than -50., the output value is set to 1.e-22.

Square. Multiplies each data value by itself.

Square Root. Sets the output value equal to the square root. For input data values less than 0.0 the output is set to -1.e22.

Inverse. Sets the output value to 1./input. For input data values of 0.0, the output is set to -1.e22.

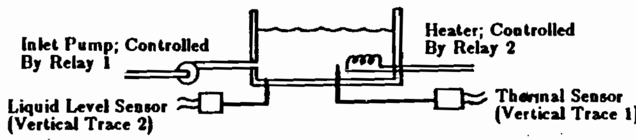
On-Lin rocessing

On line processing is provided for controllers, control of switches, coordinated stepping and sampling, and linear conversion. One of these may be selected from the User Processing block of the Setup Mode menu.

Relay Control*

Specify the on-off condition of 4 digital output lines for each of 5 "States". The transition from state to state is made when trigger levels are crossed. The trigger source can be time or any of the Vertical Trace inputs. For vertical trace triggers the next state starts when the signal crosses a level with particular slope. For time triggers, the next state starts when the specified time has passed. The sketch below indicates the operation; Section 9 of the Guide give more details.

Example: Fill tank to known height, heat contents until they reach specified temperature.




```
+User Processing<relay control>
St  Src  Slp    Lev?     #1  #2  #3  #4
A   Time  +    .00E00  off off off off
B   Time  +   5.00E00  on  off off off
C   Tr#2  +   1.60E00  off on  off off
D   Tr#1  +   4.90E00  off off off off
E   Time  +   4.90E01  off off off off
```

Controller*

Allows for open-loop or proportional control using the D/A output. The command and feedback signals can come from any of the vertical traces and the output signal can go to either of the two D/A converters. For the proportional controller the gain is specified by the user (use Delete key and Ctrl/U for editing the gain number).

Example of Controller Setup Region

```
          +User Processing<Controller>        Controller
               ^    +---------------------+   Output
         !Tr#1 / \  |!Type <Proportional  >|D/A#1
         ----|+  |-|!Gain? 1.00E+01       |---->
Command      \-/   |                      |
Signal        |    +----------------------+
            !Tr#2
                       Feedback Signal
```

*This feature works properly only with the sampling strategy "Sample/Proces/Plot Each Point".

Linear Conversl

Allows each SAMPLED vertical trace and the horizontal trace to be converted according to the slope and intercept entered in the menu. The slope and intercept are entered by the keyboard rather than being toggled; use the del key and Ctrl/U for editing. File Data is not affected by this function; to convert data already saved use the Conversion Utility.(see refp8)

Traversing*

This function coordinates sampling with the change of position of a stepper motor or a "table" driven by the D/A converter.

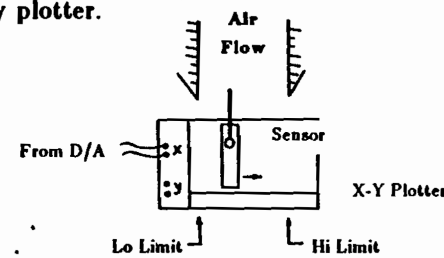
Stepper Motors Two types of stepper motors are supported. "Back/Forth" for a motor where Digital Out 1 makes it go forward and Digital Out 2 makes it go backwards. "Step/Drct" for a motor where Digital Out 1 makes it move with the direction set by Digital Out 2.

A "sense" signal can be used to protect the device from going outside the bounds set as high and low limits. Also, the device can be reset to the low limit or the high limit at the start of a scan.

D/A Driven Tables The D/A output is used as an input to a servo positioning device. The device is assumed to respond faithfully so no sense signal is used.

Parameters The number of samples each step defines how many data values are taken before the next step(s) are taken. The system can store each data value or can save only the last sample at each step. The number of steps per group determines how quickly the position will change.

Example: Traversing of Sensor Attached to x-stage of x-y plotter.




```
+User Processing<Traversing   >

!Stepper Type<d/a port 0> !Sense<none>
!Lo Limit< 5.00E-01>!Hi Limit< 3.50E+00>
!reset<lo>!Steps(bits)/grp<  4>!Dirct<+>
!# in group<  8>!Save< All>
```

*This feature works properly only with the sampling stragegy "Sample/Proces/Plot Each Point".

UnkelScope[tm] Reference Sheet

Level 2

In Program Processing

Data stored in disk files can be processed by UnkelScope Level 2. In Program processing is selected by moving to the UTILITIES choice of the command mode and pressing the down arrow key. The Utility command mode menu will appear, with the choices indicating the types of processing that can be done. A brief description of each is given below and the reference to additional information provided.

Calibration
Determines the calibration curve for a set of transducer data stored in a particular format. See Sheet Refp8 and Section 9 of the Guide.

Conversion
Allows a linear conversion of each portion of a data file. See Sheet Refp8 and Section 9 of the Guide.

Point Plot
Allows ou to enter by the keyboard a set of points you wish to plot. See Sheet Refp8 and Section 10 of the Guide.

Statistical
Allows digital filtering (high, low, band pass) and FFT related (Power Spectral Density) operations on the data. See Sheet Refp9 and Section 11 of the Guide.

Functions
Allows algebraic and other functions to be applied to the data. Included are integration, differentiation, squares, square roots, inverses, natural logs. See Sheet Refp10 and Section 12 of the Guide.

Common Features of Utilities

All Utilities act on file data, not the data currently in memory. With the exception of Point Plot you must already have a stored file before you use the utility. The input and output files cannot be the same; however the program does not check whether you are writing over an existing file.

< > indicates a toggled input; ? indicates an alphanumeric input. When typing alphanumeric inputs, use the delete key to delete one character at a time and Ctrl/U to delete the entire entry. When numbers are requested as alphanumeric input you have 9 characters to represent the number.

Calibration

This processing option allows you to take a calibration file (prepared as described in Section 9 of the Guide) and find a curvefit to the results. You must already have taken and stored the calibration data before entering this option.

You will need to enter the values from the standard you are calibrating against. The output of the process will be the conversion equation and a file that contains the original data (rearranged) and a graphical representation of the curvefit line. You can display the calibration data and compare the fitted line against the data.

When displaying the results, use <Anlg #>(or file) as the source for the Horizontal Trace. Use <Fil Ch 1> as the source for Vertical Trace 1 (this is the calibration data) and <Fil Ch 2> as the source for Vertical Trace 2 (this will be the curvefit line).

Conversion

This choice allows you to convert each component of a data file according to a linear conversion curve. For each valid channel you supply a label and the slope and intercept for the conversion. See description below about the structure of the data files.

Note also that you can perform linear conversions as an on-line processing option.

Point Plot

This option allows you to enter a set of point locations by the keyboard and plot them. You specify a file label, axis labels and up to 20 pairs of data values. The symbols are either squares or triangles. The size of the symbols is fixed rather than adjusted to be a certain fraction of the full scale range of the plot. To get "normal" size symbols, use a symbol size of about 1/20th of the range you wish to plot the data. Or chose the symbol size to represent the precision of the measured data or the analysis.

Structure of Data Files

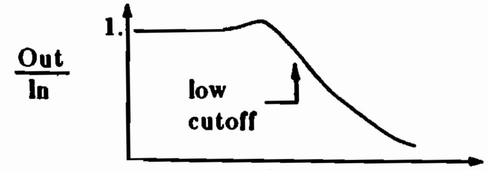
Each valid trace has 3 sets of data points: the vertical values, the time values and the horizontal values. The horizontal trace values are pertinent when the signal was plotted as a function of a sampled data channel rather than as a function of time. If the data was displayed as a function of time, the horizontal values are not relevant.

Statistical Processing

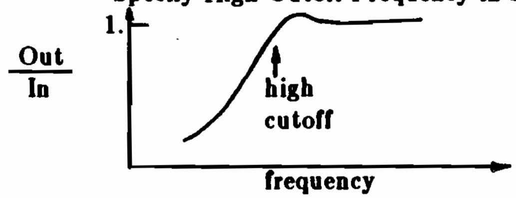
(Minimum, Maximum, Average also reported)

Filters
(2nd Order Butterworth)

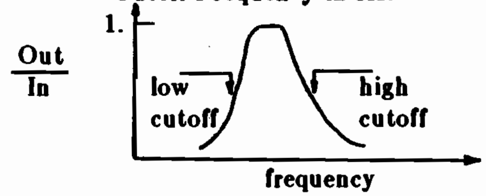
Low Pass
Specify Low Cutoff Frequency in Hz.



High Pass
Specify High Cutoff Frequency in Hz.



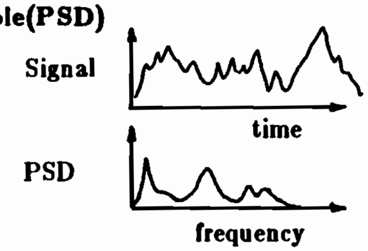
Band Pass
Specify Low Cutoff Frequency and High Cutoff Frequency in Hz.



FFT Related

Power Spectral Density (PSD)
No inputs. DC is subtracted from the signal and the PSD is scaled such that the peak value is 1. Results are stored with the PSD values in the vertical trace and the frequency in the HORIZONTAL trace. To display results choose <Anlg 0>(or file) as the source of the horizontal trace.

Example(PSD)



Unkel Software Inc. — UnkelScope

UnkelScope Quick Reference to Main Menu

QUIT
Exit program - lose any data not yet stored on the disk.

SETUP MODE
Define Parameters of data sampling and data display Data Stored in Files is retrieved by toggling Trace 1 Source to "Fil Tr 1" for example.
(Section 4.4)

SAMPL/DSPLY
Starts and controls sampling and display "Magic" when in Single sweep mode and program has just finished a sweep:
Ctrl/E - enter Edit Data
Ctrl/F - make a hardcopy
(Section 4.5)

GET SETUP
Retrieve a set of SETUP MODE conditions earlier saved with SAVE SETUP.
(Section 6.2)

SAVE SETUP
Save the current SETUP MODE parameters. This does NOT save the data itself.
(Section 6.2)

SAVE DATA
Save the current data on the disk. Valid data in Traces 1,2,3, and 4 will be stored in one file. Valid data from Traces 5,6,7, and 8 will be saved in a Second File.
(Section 4.7)

PRNT SETUP
Makes a hardcopy of the Setup Paramters.
(Section 4.6)

PRNT DSPLY
Redraws the SMPL/DSPLY screen and then makes a hardcopy of it.
(Section 4.6)

UTILITIES
A variety of functions most of which operate on previously stored data files.
See Sub-Menu below.

File Name Conventions:
UnkelScope allows only 10 characters for a filename. You must also conform to DOS conventions. We suggest names of the form
filnam.ext
Use no embedded blanks, and use only letters and numbers.

User Processing: (selected in the Setup Menu

Relay Sequencing - Use Digital output to turn devices on/off at specific times or analog input voltage levels. (Section 13.1)

Process Controllers - Open Loop, proportional, and PID Controllers can be selected. (Section 13.2)

Traversing - Use this option to "move" a device in coordination with sampling data.
(Section 13.3)

Linear Conversion - Vertical Traces 1,2,3,and 4 and either Time or Horizontal Trace 1 can be altered by a Linear Conversion.
(Section 14.0)

Synchronous Averaging - Successive scans of Traces 1,2,3 and 4 will be averaged and stored with this option (Section 14.1)

Unkel Software Inc.

UnkelScope

UnkelScope Utilities

Return Scope
Return to
Main Menu

Calibration
of transducer. Data must be taken in format of Ch. 9.

Conversion
(linear) of stored data
(Ch. 9)

Point Plot
Keyboard entry of data points to plot. (Ch.10)

Statistical
- Low Pass Filter
- High Pass Filter
- Band Pass Filter
- Power Spectrum

(Ch. 11)

Functions
- Integration
- Differentiation
- Subtract average
- Absolute value
- Peak from start
- Peak from end
- Floor(minimum)
- Ceiling(maximum)
- Natural Log
- Exponential
- Square
- Square Root
- Inverse

(Ch. 12)

Edit Data
Scroll cursor thru currently displayed data to get numerical values of the points.
(Ch. 15)

Make Procedure **Get Procedure**
Procedures (macros) allow you to automate a sequence of data taking, display and processing steps. Make procedure starts UnkelScope remembering your keystrokes until you enter Make Procedure a second time. Get procedure allows you to re-run a previously made procedure and to combine procedures together. Ctrl/P allows you to abort a procedure in progress.

Facts About
Shows min, max, avg, rms fluctuation and does a linear curvefit of the data contined in each trace of a file.
(Ch. 17)

"Pass option" If you specify the name = as the output of a Utility, no file will be written. You can then specify = as the input to the next step, and make a sequence where the intermediate steps are not saved - this saves time and disk space. You must not exit the Utilities mode before saving the last step of processing.

Unkel Software Inc.

62 Bridge Street, Lexington, MA 02173

Summary of Graphical Editing Features

To Activate Graphical Editing:

With All UnkelScope Products. Sample with Single Sweep Option, Press Ctrl/E when the message "Press A to Scan Again...." appears. Or Display previously stored data only with SMPL/DSPLY and press Ctrl/E when the message "Press Y to return to ..." appears.
With Level 2+. Move to the Edit Data Choice of the Utilities Menu.

To Return from Graphical Editing:

If you are moving the cursors, press ESC to return to the Edit Data Menu, then toggle to get 'Return-Scope' and press return. If you are in the Cursor parameter selection menu, press Home then toggle to get 'Return-Scope' and press return.

Cursor Motion Commands.

First Point (Home)	Other Cursor	+shft:Prev Peak +Ctrl:Next Peak
<-- +Ctrl:4<-- +Shft:32<-		--> +Ctrl:4--> +Shft:32->
Last Point (End)	Other Cursor	+shft:Prev Minimum +Ctrl:Next Minimum

Many of these features are available only in Releases 2.10 and later

Functions while moving Cursors

Function Keys BY THEMSELVES

1 Save a File with info between Cursors	F2
F3 Open a file for output of message line	F4 Close the file for message line output
F5 Integrate area between curve and Cursors	F6 Linear Curve-Fit; slope & intercept to message line
F7 Enter Label to message line	F8 Current Vert cursor value to message line
F9 Next Panel	F10 Previous Panel

Function Keys WITH THE SHIFT KEY DOWN

F1	F2
F3 Store the message line into the output file	F4 Clear the message line
F5 Minimum & Maximum to the message line	F6
F7	F8 Current Horz cursor value to message line
F9 Expand by 2 about the active cursor	F10 Contract by 2 about the active cursor

Function Keys WITH THE CTRL KEY DOWN

F1	F2
F3 Print the message line to the printer	F4
F5 Average & RMS fluct. to message line	F6
F7	F8 Current Time cursor value to message line
F9 Return display to the default settings	F10

(617) 861-0181

EditMacro Capabilities

DO NOT: Call one EditMacro from within another or from within a Procedure.

Function Keys WITH THE ALT KEY DOWN

F1	F2
Save file with current EditMacros	Retrieve file with EditMacros
F3	**F4**
To start saving an EditMacro Press ALT/F3 followed by the EditMacro you wish to save	End the Macro by typing ALT/F4
F5	**F6**
Execute the currently stored ALT/F5 EditMacro	Execute the currently stored ALT/F6 EditMacro
F7	**F8**
Execute the currently stored ALT/F7 EditMacro	Execute the currently stored ALT/F8 EditMacro
F9	**F10**
Execute the currently stored ALT/F9 EditMacro	Execute the currently stored ALT/F10 EditMacro

Unkel Software Inc.

(617) 861-0181 telex: 9102409722

62 Bridge Street, Lexington, MA 02173

Quick Start Guide for the IBM DACA Board Release 3.05

This Guide may allow you to quickly install and run UnkelScope with your data acquisition system. They assume that you have a reasonable familiarity with the computer and data acquisition board. The Installation Guide and Users Guide give full details.

1. Select the proper UnkelScope Program Diskette.

Version	Conditions	Program Disk	Special Instructions
Junior*	Less than 640k bytes	PC-JR-384	follow 384k instructions
Level 1	"	PC-L1-384	"
Level 2+	"	PC-2P-384	"
Level 1	640k bytes or more	PC-L1-640	follow 640k instructions
Level 2+	"	PC-2P-640	"

*Only the 384k program disk is provided with UnkelScope Junior.

2. Make the working copy of UnkelScope.

<u>IF YOU HAVE A HARD DISK SYSTEM:</u>

a. Make a subdirectory USCOPE. Type: **mkdir \uscope**
b. Move to that subdirectory. Type: **cd \uscope**
c. Put the Program disk into Drive A:
   Copy files from floppy to Hard Disk. Type: **copy a:*.* c:**
d. Put the disk labelled PC-SUP into Drive A:
   Copy files from floppy to Hard Disk. Type: **copy a:*.* c:**
e. Copy file from the appropriate subdirectory,
   Ie. XXX is 384 or 640 as defined above. Type: **copy a:\XXXk c:**
f. IF YOU HAVE A HERCULES GRAPHICS CARD (or Equal.) Type: **copy a:\herc c:**
g. Return the diskettes to their jackets.

<u>IF YOU HAVE A FLOPPY DISK SYSTEM:</u>

a. Copy the Master Program Diskette to a Diskette that will be used as the working copy
   i. To the DOS prompt, Type: **diskcopy a: b:**
   ii. Insert the Program Diskette in Drive A:.
   iii. Insert a **blank** diskette into Drive B:.
   iv. Press Enter.
   v. Answer "n" when prompted whether to copy additional disks.
b. Return the Master Program Diskette to its jacket.
c. Label the Working Program diskette and put it into Drive A:.
d. Place the Disk labelled PC-SUP into Drive B:.

3. Configure the Board and the Software.
   Run the UnkelScope Configuration Program.
   <u>IF YOU HAVE A HARD DISK SYSTEM</u>, Type: **configur**
   <u>IF YOU HAVE A FLOPPY DISK SYSTEM</u>, to the A> prompt, Type: **b:configur**
   Follow the instructions. When all selection are correct select the menu option "EXIT - WRITE FILE" to write the configuration information.

4. Install the data acquisition system hardware.

5. Restart the computer and get it ready to run UnkelScope.

   These instructions must be followed each time you reboot your computer. Use the DOS AUTOEXEC.BAT file to automate some or all of these steps.

IF YOU HAVE A HARD DISK SYSTEM

If you follow these instructions, the copy of UnkelScope in the \uscope directory can be used by all users and the data files will be stored and retrieved from the current directory.

a. Set the Path to include \uscope. Type: **path=c:\uscope**
b. Make a subdirectory from which to work:eg. Type: **mkdir \SMITH**
c. Move to the subdirectory. Type: **cd \SMITH**
d. IF YOU HAVE A HERCULES GRAPHICS CARD Type: **hgc full**
   int10
e. IF YOU HAVE A COLOR GRAPHICS ADAPTER, move to the directory with file GRAFTABL.COM and Type: **graftabl**
f. IF YOU HAVE AN 80286 or 80386 BASED COMPUTER AND DO NOT HAVE A MATH COPROCESSOR, Type: **set no87=0**

IF YOU HAVE A FLOPPY DISK SYSTEM

If you follow these instructions, your data files will be stored and retrieved from the diskette in Drive B:.

a. IF YOU HAVE A COLOR GRAPHICS ADAPTER
   Insert your DOS diskette into Drive A: and Type:**graftabl**
b. IF YOU HAVE A HERCULES GRAPHICS CARD, insert the Disk labelled PC-SUP into Drive B:, Type: **b:\herc\hgc full**
   b:\herc\int10
c. IF YOU HAVE AN 80286 or 80386 BASED COMPUTER AND DO NOT HAVE A COPROCESSOR, Type: **set no87=0**
d. Put the Working copy of UnkelScope into Drive A:
e. Put a Formatted diskette into Drive B:
f. Set the Path to include Drive A: Type: **path=a:**
g. Set Drive B: as the default disk drive. Type: **b:**

6. Connect a signal to the data acquisition system.
   If available connect a signal to the first A/D channel.

7. Run UnkelScope. Type: **scope**

   The Short Guide, Users Guide and Installation Guide provide details.
   If you have difficulties, give us a call. Or, if your problem is complex or you are overseas, send us a FAX.
   Phone Service: (617)861-0181
   FAX: (617)861-1850

8. Software Registration. UnkelScope Level 1 and 2+ come with 1 year of service and software updates. To be receive the updates you must return the registration forms located in the back of the documention notebook.

Unkel Software Inc.

(617) 861-0181 telex: 9102409722

62 Bridge Street, Lexington, MA 02173

UnkelScope for IBM DACA Boards: Release 3.05 Notes

1.0 DISKS PROVIDED (Release 3.05)

The distribution diskettes are identified in the upper right corner. Check that you have received the correct diskettes for your Version:

PC-XX-384 Release 3.05, PC-XX-640 Release 3.05 (not provided with UnkelScope Junior), PC-SUP Release 3.05, UTI Release 3.01, TUT Release 1.00

where XX is either JR, L1 or 2P depending on whether you purchased Junior, Level 1 or Level 2+ respectively.

All versions of UnkelScope will use a math coprocessor if it is present, but a coprocessor is not required.

2.0 DOCUMENTATION

2.1 QuickStart Guide (Release 2.07 and Later)

This guide is provided to help the relatively experienced user get UnkelScope installed and running. For more detailed descriptions use the UnkelScope Installation Guide and the UnkelScope Users Guide.

2.2 Tutorial Disk (Release 3.01)

A tutorial has been included, but it is Release 1.00 and is prepared in our "old screens." You may find it useful, but it will look different from the program. This program will work only with IBM Graphics or equivalent.

3.0 INSTALLATION INSTRUCTIONS (Release 2.01 and Later)

If you are installing the software (and hardware) you should read and follow the instructions of either the QuickStart Guide or the full Installation Guide.

4.0 MISCELLANEOUS

4.1 AT (or compatible) Without a Math Coprocessor (All Releases)

If you have an 80286 or 80386 based computer and do not have a math coprocessor installed, you must Type:

set no87=0

4.2 Graphics Support

UnkelScope now supports CGA, Hercules and EGA graphics systems. Be sure to read the instructions (Installation Guide or QuickStart Guide) for any special procedures for your graphics system.

4.3 Graphic Editing Features

Our customers have found the Graphic Editing features of UnkelScope very useful. You may wish to 'browse' through this section of the Users Guide to see how to activate these features for your version of UnkelScope.

4.4 Modifications to the Triggering Block (Release 3.01 and Later)

In anticipation of some functional changes, we have made some cosmetic changes in our Triggering Block. For Release 3.05 the functions are the same as described in the documentation.

5.0 OFFLINE UTILITIES PROGRAMS

Release 3.00 and later save data files in a different format than described in Chapter 7 of the Users Guide. This will not concern you unless you wish to translate files into ASCII Files, Lotus ".PRN" files, or wish to write programs that access our files directly. The new files are written to be compatible with a powerful matrix manipulation program PC-MATLAB. The Document "File Structure and File Format Translation for UnkelScope" found in Chapter 7 describes the file format and describes the program "UTRANS" which you can use to translate 3.00 or later files into ASCII or Lotus ".PRN" Files. The structure of the ASCII and Lotus Files is the same as described in the Users Guide. If you wish to access our data files directly from a program that you are writing, please call us and we will be happy to assist you.

6.0 IBM DACA SPECIFIC

6.1 There is no error checking for clock faults, so you must insure that the sampling rate selected can be handled by your hardware set.

6.2 Problem with the horizontal trace has been fixed. (2.05 and greater)

Configuration, Installation and Checkout of

UnkelScope®

for the IBM DACA Board

Unkel Software, Inc.
62 Bridge Street
Lexington, MA 02173
(617) 861-0181

Table of Contents

Table of Contents

1. Overview

Installing the UnkelScope System consists of:

a) making a working copy of UnkelScope from the factory diskettes;

b) running the configuration program to identify the components in your computer system and any hardware specific options in your data acquisition system;

c) checking or setting any hardware switches/jumpers and making any necessary connections; and,

d) checking that the software and hardware are correctly installed.

This guide is written with the assumption that you are installing the hardware and the software together. If you have already installed the hardware you may be able to skip certain discussions.

If you are already familiar with the hardware in your data acquisition system, the UnkelScope QuickStart Notes may provide enough detail to install and start using the software.

The hardware manual that came with your data acquisition system will provide you with more details, but for your convenience the configuration choices are discussed in Sections 3 and 4, along with instructions on how to set any switches/jumpers on the board. In most cases this guide will provide all the information you need to install and start using UnkelScope and the data acquisition system.

2. Cautions

Computer hardware and in particular the data acquisition board require more care than some analog instruments. The most significant precautions are listed here, but you should read the computer manuals and computer peripheral manuals to make sure you know the correct operation procedures.

A) You MUST switch off the main power to the computer when you install or remove a circuit board (such as the data acquisition board) from the computer. If you remove or insert the board with the power on, DAMAGE may result from sudden voltages applied to the board.

B) You MUST NOT apply high voltages to any of the lines of the data acquisition board. The maximum voltage for the IBM DACA Board Analog inputs is +/- 20 volts. Higher voltages will damage components on the board. Further, the board is internal to the computer and high voltages may damage the computer as well.

C) You SHOULD make a backup copy of software you receive.

D) The Digital input and output lines should be connected only to TTL compatible devices. Refer to the IBM DACA documentation if you are not familiar with TTL logic devices.

3. Review of IBM DACA Board Choices

Hardware "jumpers" and switches are used to define the configuration of the data acquisition board. The board comes with factory jumper settings for each of the options, but in some cases you may wish to alter these. In any case, you should understand what the issues are. The choices are grouped into three categories: hardware options specific to the board as purchased; user configured choices relating to the analog and digital inputs and outputs; and user configured choices for communication with the computer. Further, to get the best performance over a range of possible computer types and configurations, UnkelScope needs to know some details about your computer. This section discusses all of these choices.

The jumpers and switches must be set while the board is removed from the computer. If you are installing the software and hardware together, make all of the configuration selections and run the configuration program before installing the board. The configuration program will provide a jumper summary, detailing which jumpers should be installed or removed. If you are unsure of which option to select, select the default or the factory setting, and alter it later if necessary.

3.1 Hardware Features of the IBM DACA Board

The IBM DACA has 4 differential analog inputs (A/D channels), 16 digital input lines, 16 digital output lines, and two analog output (D/A channels). Expansion boards may be available for the IBM DACA, but these are not supported by UnkelScope and their operation is not discussed here. Support of 2 DACA boards concurrently may be available in Releases beyond Release 3.05.

3.2 User Configured Choices for Analog and Digital Input and Output

These selections control how the board interacts with the analog and digital signals you connect. Specifically, Section 3.2.1 discusses the choices for the voltage range of the analog to digital inputs and Section 3.2.2 discusses the choices for the voltage range of the digital to analog outputs.

3.2.1 Analog Input Range

The choices for analog input range are -10 to 10 volts, -5 to 5 volts and 0 to 10 volts. With the unipolar setting only positive voltages can be measured; with the bipolar setting positive and negative values can be measured. The resolving power of the analog-to-digital converter is given as the full scale range divided by 4096 for this 12 bit converter. By limiting the signal range to positive values only, a factor of 2 better resolving power is obtained; of course, if the signal is not of the polarity you expect, the device will return a zero value. In cases where you are using close to the full range of the device, the resolution (even for the 12 bit board) is about 0.025% which is satisfactory for most purposes. Thus, the factor of 2 is often not significant.

Unkel Software suggests that you use the bipolar configuration, unless you are very sure you will have no signals which change sign. Whatever range you choose, NEVER PUT IN VOLTAGES ABOVE THE RECOMMENDED LEVEL.

3.2.2 Digital to Analog Output Range

The two digital to analog (D/A) converters can be configured to have output ranges of -10 to 10 volts, -5 to 5 volts, or 0 to 10 volts. It is usually convenient (but not required) to chose a bipolar D/A range if the A/D is bipolar and a unipolar D/A range if the A/D is unipolar. For a complete discussion of the options for the D/A, read the IBM DACA manual.

3.3 Choices for Communication With The Computer

For the IBM DACA boards a board number must be selected. This selection determines how the computer communicates with the board. For the software to work properly, it must know what was selected.

The board number determines which Input/Output (I/O) Ports the board uses. Every peripheral (such as a printer, plotter, or data acquisition board) communicates through the I/O ports. When selecting the board number, make sure that the I/O ports the board uses do not conflict with other peripherals in the computer. If all you have is a monitor, a printer, and a plotter, then any board number will do. If you have many devices on the computer, such as a mouse, other data acquisition boards, etc. you must check that all devices in the computer use different I/O ports. The I/O port numbers used for each possible selection of the IBM DACA adapter number are listed below.

Hex Addresses Used By the IBM DACA Board

	Adapter Number			
	0	1	2	3
port 0	2E2	6E2	AE2	EE2
port 1	12E2	16E2	1AE2	1EE2
port 2	22E2	26E2	2AE2	2EE2
port 3	32E2	36E2	3AE2	3EE2
port 4	42E2	46E2	4AE2	4EE2
port 5	52E2	56E2	5AE2	5EE2
port 6	62E2	66E2	6AE2	6EE2
port 7	72E2	76E2	7AE2	7EE2
port 8	82E2	86E2	8AE2	8EE2
port 9	92E2	96E2	9AE2	9EE2
port 10	A2E2	A6E2	AAE2	AEE2
port 11	B2E2	B6E2	BAE2	BEE2
port 12	C2E2	C6E2	CAE2	CEE2
port 13	D2E2	D6E2	DAE2	DEE2
port 14	E2E2	E6E2	EAE2	EEE2
port 15	F2E2	F6E2	FAE2	FEE2

Interrupts are used by some programs to allow the data acquisition board to let the computer know when it has completed a particular task. The UnkelScope program does not use the interrupt capability of the IBM DACA board. However, you must still chose an interrupt request line that does not conflict with any other board. Valid choices for the interrupt request line are 3, 4, 5, 6, and 7.

4. Review of Computer Hardware Configuration Choices

While the software interface is independent of the computer hardware, the performance of the system is not. To allow UnkelScope to perform best with your hardware you must respond correctly to the hardware option selections in the configuration program. In addition, UnkelScope can be configured to take more samples on fewer traces. This selection is discussed in Section 4.5.

4.1 Computer Type

The processor speed affects both the maximum sampling rate and the maximum rate at which data can be plotted on the screen while it is being acquired. For example, the IBM PC/AT may plot data faster than the IBM PC. Select the system that best matches your computer system. If you have a compatible not listed and are not sure which computer it best matches, select the IBM PC option for a start and consult Unkel Software for advice. If the processor speed is of particular importance, the configuration program will print a message telling you the processor speed it will use. If this speed does not agree with the speed of your processor, call Unkel Software.

4.2 Math Coprocessor

The math coprocessor can speed up the plotting of points on the graphics screen. Certain versions of UnkelScope may require the math coprocessor; this will be indicated on the Distribution Disk, in the Release Notes, and in the QuickStart Notes.

4.3 Graphics Support

The sections below detail the Graphics Adapter/Monitor combinations supported by UnkelScope and describe any special setup procedures required.

A. CGA (Color Graphics Adapter) with a Color (or compatible monochrome) Monitor or equivalent. This gives a screen size of 640 X 200 with pixels either on or off (i.e. 1 color). There are two CGA options depending on whether you have a Monochrome or Color Monitor. If you have a Monochrome Monitor you must select the CGA and Mono Monitor Option.

   Special Setup Procedure. To get reverse video to work properly in the Graphic Editing menu, you must load the extended graphics character set using the DOS program GRAFTABL.COM. Since this must be done each time you reboot your system, you may wish to put the command in the AUTOEXEC.BAT file for your system. To load the graphics character set, move to the directory with the DOS program GRAFTABL.COM, type: **graftabl**, and press enter.

B. Hercules Monochrome Graphics Card with a Monochrome Monitor, or equivalent. This gives a screen size of 720 X 348, with pixels either on or off (i.e. 1 color).

Special Setup Procedure. For Hercules emulation to work properly, you must run two programs before you run UnkelScope. Since this must be done each time the system is rebooted, you may wish to include these steps in your AUTOEXEC.BAT file. The first of these two steps is to activate the Graphics mode of the Hercules card. For a Hercules Graphics Card or a true Hercules compatible, this is done by running the program HGC.COM provided in subdirectory \herc on the Utilities diskette or the Supplementary Files diskette provided with UnkelScope. Specifically, you type: **hgc full** and press enter. For some Hercules compatible boards this first step may be different. Check your documentation for specific instructions. The second step is to load the Interrupts that relate to graphics. This is done by running the program INT10.COM provided in subdirectory \herc on the Utilities diskette or the Supplementary Files diskette provided with UnkelScope. Specifically, you type **int10** and press enter.

C. EGA Adapter with Monochrome Monitor or equivalent. This gives a screen size of 640 X 350, with pixels either on or off (i.e. 1 color).

Special Setup Procedures. You may have to run a configuration program provided with your EGA Adapter to activate this mode if you have not chosen it as the default mode on your EGA Adapter. You should consult the documentation for your EGA Adapter.

D. EGA Adapter with Hi-Res Monitor or equivalent. This gives a screen size of 640 X 350 and uses different colors for the two traces shown on the screen. There are two selections depending on how much memory your EGA card has. If your EGA card has only 64 kbytes, you must select the EGA Hi-Res (64 kbyte) option or Trace 2 may not be plotted properly.

E. EGA Adapter with Color Monitor or equivalent. This gives a screen size of 640 X 200 and uses different colors for the two traces shown on the screen.

Consult Unkel Software if you have any questions about compatibility.

4.4 Printer Type

UnkelScope will work with any IBM graphics compatible printer. This includes a variety of Epson printers, Okidata printers, the IBM Pro Printer and others. Certain Laser Printers and 24 pin printers may require special emulation software to work with UnkelScope. Some IBM compatible printers allow you to switch off the "Special Graphics Characters." If you get unusual characters where the borders to the blocks normally are, you need to either:

a) change the Jumper Setting on your printer to select the Special Graphics Character Set (IBM Graphics Compatibility); or,

b) run a program, supplied with your printer, that loads or sets the Graphics Character Set into the Printer. Consult your printer documentation for details on how to do this.

4.5 Number of Traces for UnkelScope

UnkelScope can acquire data on up to 8 traces (4 traces for Junior), but may be configured to take additional samples on each trace by allowing fewer traces. The details are summarized on the table below. Note also that selection of the 2 trace or 1 trace option will mean that you cannot perform "cross-plots" (x vs. y plots). The constraints on sample size come from a variety of sources, so the product of traces times maximum samples on each will not be the same.

Number of Traces	Maximum Samples on Each: 384 kbyte version	Maximum Samples on Each: 640 kbyte version	Cross-Plot
8 (4 Junior)	1024	4096	yes
2	2048	8192	no
1	8192	16,384	no

5. Configuring UnkelScope and Your Data Acquisition System

If you have already configured or installed your data acquisition hardware, you should fill in Table 5.1 on page 5-2 and run the UnkelScope Configuration program. (Go to Section 5.3.) If you have not already configured the board, you should fill in Table 5.1 as you run the Configuration program and when done make any necessary changes to the jumper or switch settings that may be on your system hardware.

5.1 Summary and Table of the Board Configuration

Section 3 discussed the Board configuration choices; the data acquisition board manual also discusses these details. Table 5.1 summarizes these choices and gives you a place to write your selections. If you know your choices, write them in now. If not, you can write them in as you go through the configuration program.

Section 4 discussed the computer hardware configuration choices. Table 5.1 also contains space for recording your computer configuration. This information is passed from the configuration program to the UnkelScope program for proper use of the hardware.

5.2 Making a Working Copy of UnkelScope

We suggest that you make a copy of all disks at this time. Never use the original disks to run UnkelScope; use the copy or subdirectory made with the instructions in this Section. The original disk should have a write protect tab on it to prevent you from accidentally writing over it. The configuration program writes a disk file at the end and therefore cannot be run with a write protect tab.

Selecting the Proper UnkelScope Program Diskette. You must select the proper program diskette to suit your computer hardware.

Version	Conditions	Program Disk	Special Instructions
Junior*	Less than 640k bytes	PC-JR-384	Follow 384k instructions
Level 1	"	PC-L1-384	"
Level 2+	"	PC-2P-384	"
Level 1	640 kbytes or more	PC-L1-640	Follow 640k instructions
Level 2+	"	PC-2P-640	"

*Only the 384k disk is provided with UnkelScope Junior.

If you have a hard disk system, follow the instructions in Section 5.2.1; if you have no hard disk follow the instructions in Section 5.2.2.

TABLE 5.1 Configuration Choices

Computer - IBM DACA interface

	your selection	factory setting
Board Number		0
Interrupt Request Line (3, 4, 5, 6, or 7)		7

Analog Input Configuration

	yours	factory
Analog Input Range: (0-10, +/-5, +/-10)		

Analog Output Configuration

	yours	factory
Analog Output Range (0-10, +/-10)		

Computer Specifications

Computer Type	
Coprocessor	
Screen Graphics Type	
Printer Type	

5.2.1 Hard Disk Systems

We suggest that you make a subdirectory specifically for UnkelScope and another subdirectory for each user of UnkelScope. By keeping individual user subdirectories, you will be able to keep your data separate from others who use the system. If you are taking many different types of experiments, you may wish to keep a separate subdirectory for each experiment. This section explains how to set up UnkelScope so that one copy of it on the hard disk will serve many users.

Boot the system so that you have the DOS prompt. Then follow the step-by-step instructions below. The characters to be typed follow the DOS prompt C> and in each case are completed by typing the enter key.

A. Make a subdirectory called "uscope" for the UnkelScope Program. Type:

C> **mkdir \uscope**

B. Move to the subdirectory called \uscope. Type:

C> **cd \uscope**

C. Place the proper UnkelScope Program Diskette in Drive A:.

D. Copy the UnkelScope Program and files into the subdirectory \uscope. Type:

C> **copy a:*.* c:**

E. Remove the floppy disk and return it to its jacket.

F. Now place the UnkelScope "Supplementary Files" diskette in Drive A:.

G. Copy certain contents on this disk into the subdirectory c:\uscope.

For all versions, Type:

C> **copy a:*.* c:**

If you are using the 640 kbyte version, Type:

C> **copy a:\640k*.* c:**

If you are using the 384 kbyte version, Type:

C> **copy a:\384k*.* c:**

H. If you have a Hercules Monochrome Graphics Card or equivalent, copy the Hercules subdirectory onto the hard disk. Type:

C> **copy a:\herc*.* c:**

I. Remove the floppy diskette from Drive A: and return it to its jacket.

J. Confirm that the necessary files were copied. Type:

C> **dir c:**

The following files should be listed:

configur.exe
ibmcan.aux
scope.exe
pclvxx.yyy where xx is 2p,jr, or l1, and yyy is 384 or 640
memory.inp
ex2se1.set
ex2se2.set
ex2se3.set
ex2se4.set
ex2se5.set

If you have a Hercules Monochrome Graphics card, the following files should be listed:

hgc.com
int10.com

The necessary programs have now been copied to the hard disk in the subdirectory \uscope. The original disks should be returned to their disk jackets in the documentation notebook.

K. Make a subdirectory for each user. This is done with the DOS "mkdir" command as in Step A. For instance if you want a subdirectory called "SMITH", Type:

C> **mkdir \smith**

To move to that directory, type:

C> **cd \smith**

Skip Section 5.2.2 and continue with Section 5.3.

5.2.2 Floppy Disk Systems

You should make a copy of the UnkelScope Program Diskette and use this copy to work from. First, boot the system so that you have the DOS system prompt. Then follow the step-by-step instructions below. The characters to be typed follow the DOS prompt A> (or B>) and in each case are completed by typing the enter key.

A. Place the proper UnkelScope Program Diskette in Drive A:. **DO NOT** remove the write protect tab.

B. Insert a blank disk into Drive B:. BE SURE THAT THE DISK CONTAINS NOTHING YOU WANT BECAUSE ALL FILES WILL BE WRITTEN OVER IN THE NEXT STEP.

C. Type:

A> **diskcopy a: b:**

Follow instructions from the monitor.

D. Remove the floppy disk in Drive A: and return it to its jacket.

E. Confirm that the necessary files were copied. Type:

A> **dir b:**

The following files should be listed:

ibmcan.aux
scope.exe
pclvxx.yyy where xx is 2p,jr, or 11, and yyy is 384 or 640
memory.inp

F. Remove and label the diskette as required by the software license.

The necessary programs have now been copied to the floppy disk. The original disk should be returned to the disk jacket in the documentation notebook.

G. Place your Working Program Disk in Drive A: and place the disk labeled PC-SUP into Drive B:.

5.3 Running the Configuration Program

The Configuration program lists for the UnkelScope program:

a) the configurations set on your data acquisition system; and,

b) some other important characteristics of your computer system.

If you are using a hard disk system, read Section 5.3.1, skip Section 5.3.2, and continue with Section 5.3.3.

If you are using a floppy disk system, skip Section 5.3.1, and read Sections 5.3.2 and 5.3.3.

5.3.1 Getting Ready for Hard Disk Systems

To prepare for the Configuration program, you need the subdirectories created in Section 5.2.1.

Move to the subdirectory \uscope of the hard disk by typing:

C> **cd c:\uscope**

5.3.2 Getting Ready for Floppy Disk Systems

To prepare for the Configuration program, you need the Working Program Disk prepared as described in Section 5.2.2.

When you have the proper disk, boot the system and get the DOS prompt.

A. Put the Working Program disk of UnkelScope into Drive A:.
B. Put the disk labelled PC-SUP into Drive B:.
C. Set Drive A: as the default disk drive.

Type: **A:**

5.3.3 Running the Configuration Program

<u>If you are using a Hard Disk System</u>, Type: **configur**
<u>If you are using a Floppy Disk System</u>, Type: **B:configur**

The screen will clear and give you directions to follow. Figure 5.1 shows the screen as it might appear. When you start, the "MODIFY CHOICES" option will appear in reverse video. To modify a selection, use the up or down arrow keys to move to the line and then use the left or right arrow keys to toggle to the selection you wish. When you have selected the desired choices for all the parameters, you should use the up and down arrow keys to move to the command line. Then use the left and right arrows to move to the next activity. To save your configuration choices, select the "EXIT - WRITE FILE" choice and press the down arrow key. This will write the file that UnkelScope reads. To exit without writing a file, select the "QUIT" option and press the down arrow key.

If your board or system has any jumpers or switches, you will be able to view their settings if the second menu line entry reads "JUMPERS/PRINT". If this is the case you can move to this selection, press enter, and the upper portion of the screen will show the jumper settings. You can obtain a printout of the screen by moving to the print choice in the sub-menu that appears. If the second option on the menu line is "PRINT" then you can obtain a printout by moving to the selection and pressing the down arrow key.

```
              Welcome to the UnkelScope(tm) Configuration Program
                                  for the
                  IBM PC Data Acquisition and Control Adapter
                     Release 3.05 (c) 1985 Unkel Software


        Move to the selection you want to change with the arrow keys
        on the right of the keyboard.  Then use the right and left
        arrow keys to "toggle" between choices.


Commands:     Modify        Print Summary   Exit-write file  Quit-no changes

 ------------- DACA Board ---------------   ------------- Computer ---------------
| A/D Voltage Range [ -5 to 5 ]         | |   Computer [IBM PC    or equal ]     |
| D/A Voltage Range [ -5 to 5 ]         | |    Printer [IBM Graphics          ]  |
|    Adapter Number [ 0 ]               | |   Graphics [IBM CGA & Mono   /Equal] |
|    Interupt Level [ 7 ]               | | Coprocessor [Not Installed ]         |
|        # Traces  Size 384k  Size 640k | ---------------------------------------
| Config [    4      1024 pts   4096 pts|
 ---------------------------------------
```

Figure 5.1 Sample Screen from the Configuration program.

6. Installing the Data Acquisition System and Review of Connections

You should now have a configured "program" diskette or a subdirectory with the results from running the configuration program. You should also have set the jumpers in place. If you have not completed these steps, please do so now.

6.1 Installing the IBM DACA Board

If the cover to the computer is not already removed, you should remove it at this time. REMEMBER to turn off the power in anticipation of inserting the board into the computer. Consult the computer operations manual or the IBM DACA hardware manual if you are unfamiliar with installing boards into the computer.

The IBM DACA Board must be installed in one of the "full" slots. Remove a vacant back plate by undoing the screw at the top. You should retain any packaging and the backplate removed.

Insert the black plastic retaining piece in the appropriate holes on the front inside of the computer. Slide the board in carefully and return the screw removed from the backplate to hold the board securely.

6.2 Connections

Figure 6.1 shows a summary of the connections to the 60 pin connector at the back of the IBM DACA board and to the screwpin connection panel.

J4 Connector Pin Locations

(Top of Computer)

Signal Name	Pin #		Signal Name
D/A Output Channel 1	1	2	D/A Output Channel 0
+ 10 Volt Reference	3	4	Analog Ground
A/D Input Channel 0 Low	5	6	Analog Input Channel 0 High
A/D Input Channel 1 Low	7	8	Analog Input Channel 1 High
A/D Input Channel 2 Low	9	10	Analog Input Channel 2 High
A/D Input Channel 3 Low	11	12	Analog Input Channel 3 High
Analog Ground	13	14	A/D CE
Digital Ground	15	16	A/D CO
Digital Input Bit 8	17	18	Digital Output Bit 8
Digital Input Bit 9	19	20	Digital Output Bit 9
Digital Input Bit 10	21	22	Digital Output Bit 10
Digital Input Bit 11	23	24	Digital Output Bit 11
Digital Input Bit 12	25	26	Digital Output Bit 12
Digital Input Bit 13	27	28	Digital Output Bit 13
Digital Input Bit 14	29	30	Digital Output Bit 14
Digital Input Bit 15	31	32	Digital Output Bit 15
Digital Input Hold	33	34	Digital Output Gate
Digital Input Bit 0	35	36	Digital Output Bit 0
Digital Input Bit 1	37	38	Digital Output Bit 1
Digital Input Bit 2	39	40	Digital Output Bit 2
Digital Input Bit 3	41	42	Digital Output Bit 3
Digital Input Bit 4	43	44	Digital Output Bit 4
Digital Input Bit 5	45	46	Digital Output Bit 5
Digital Input Bit 6	47	48	Digital Output Bit 6
Digital Input Bit 7	49	50	Digital Output Bit 7
Rate Out	51	52	Delay out
Digital Input Strobe	53	54	Digital Output Strobe
Digital Output CTS	55	56	Digital Input CTS
IRQ	57	58	Counter Output
Counter Input	59	60	Digital Ground

(Bottom of Computer)

Figure 6.1 Summary Table of Connections at the IBM DACA board 60 pin connector.

6.3 Setting or Checking the IBM DACA Jumpers and Switches

You have made your selections for the jumper settings and must now make the board conform to these settings (or must rerun the configuration program). The sections below review the jumpers or switches that have to be set and discuss the hardware. You should use Table 5.1 or the printout from the configuration program to check these settings.

To check or set the jumpers the board must be removed from the computer. If you have not yet installed the board, you should remove the board from its packaging. As a precaution against any static electricity built up, it is good practice to touch the metallic frame of the computer before taking the board out of the package.

If the board is already in the computer and you wish to check/set any of the jumpers, you should remove it from the computer at this time. REMEMBER to turn off the power to the computer when you insert or remove any device.

A sketch of the IBM DACA board is shown in Figure 6.2 and identifies the location of the hardware jumpers and switches referred to below.

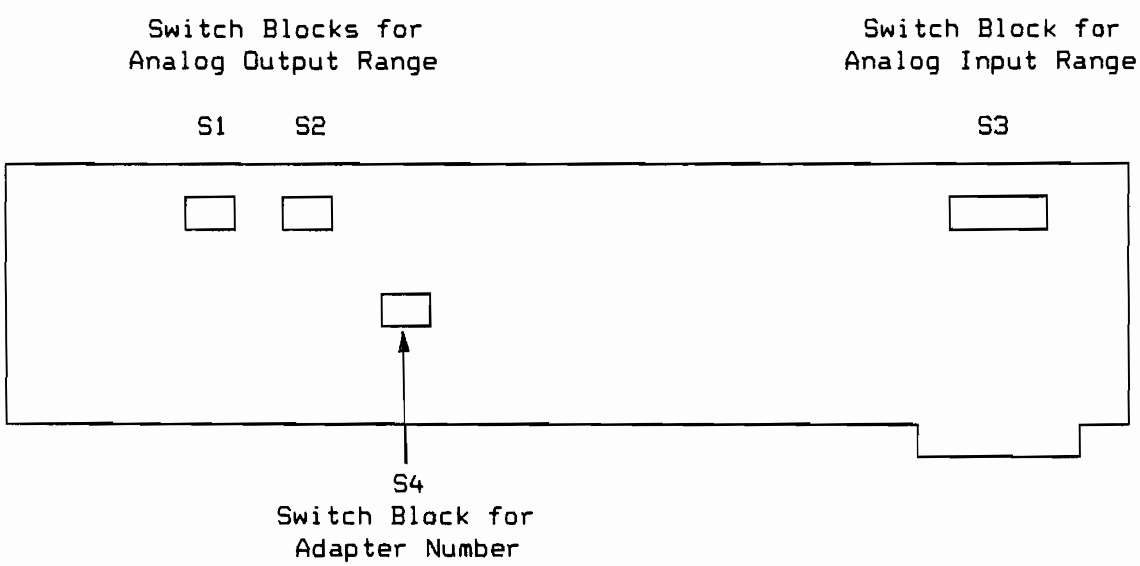


Figure 6.2 Diagram of the IBM DACA Board showing the locations of the Jumpers and Switches.

6.3.1 Adapter Number

The Adapter number is selected by setting switches 1 and 2 of Switch block S4. The choices are:

Adapter Number — Switch Block S4

0

	1	2
on		
off	X	X

1

	1	2
on	X	
off		X

2

	1	2
on		X
off	X	

3

	1	2
on	X	X
off		

6.3.2 Analog Input Range

The analog input range is selected by switches S3-1, S3-2, S3-3, and S3-4. The switch settings required for each configuration is shown below.

Input Range — Switch Block S3

-5 to 5 volts

	1	2	3	4
on			X	X
off	X	X		

-10 to 10 volts

	1	2	3	4
on		X		X
off	X		X	

0 to 10 volts

	1	2	3	4
on			X	
off	X	X		X

6.3.3 Digital to Analog Output Range

The output ranges for the two Digital to Analog outputs are is selected by the switches 1 and 2 of switch blocks S1 and S2. UnkelScope requires that both outputs have the same range. The switch configuration for each of output ranges is shown in the figures below.

Output Range	Switch Block S1	1	2	Switch Block S2	1	2
-5 to +5	on	X	X	on	X	X
	off			off		
-10 to +10	on		X	on		X
	off	X		off	X	
0 to 10	on	X		on	X	
	off		X	off		X

7. Checking Out UnkelScope and the Data Acquisition System

The easiest way to check out the system is to run UnkelScope. Then if there is a problem, Section 8 will help you to identify it. If you do not detect any serious problems, run the UnkelScope software with the instructions in the UnkelScope Users Guide. Completing Sections 1 through 4 will finish the job of finding problems in the sampling and display portions of the software. For Level 1 and Level 2+ software, operation of the Digital I/O and the Digital-to-Analog converter is covered in later Sections of the UnkelScope Users Guide.

7.1 Equipment Needed

To make the quick test you should have a steady voltage source of magnitude about half the full scale you selected for the analog inputs, but not exceeding 4.5 volts. If you don't have such a source, complete the test below anyway.

You will also need a pair of scissors, some tape and a pen.

7.2 Quick Checkout

Boot the system and connect the voltage source to give a positive input to the first channel of analog input.

7.2.1 Signal Connections

The signal connections to your data acquisition system have been summarized in Section 6 of this guide. The connections are also summarized in the Short Guide and in a table in Section 4 of the UnkelScope Users Guide.

7.2.2 Additional Setup Requirements

Make sure your printer is hooked up and is on line.

Hard Disk System. You should already have completed the steps in Section 5; if you have not, do so now. The instructions that follow must be completed every time you reboot your computer. Use the DOS AUTOEXEC.BAT file to automate some or all of these steps. If you follow these instructions, the copy of UnkelScope in the \uscope directory serve all users and the data files will be stored and retrieved from the current directory.

A. Set the Path to include \uscope.

Type: **path=c:\uscope**

B. Make a subdirectory from which to work, for example:

Type: **mkdir \smith**

C. Move to the subdirectory.

Type: **cd \smith**

D1. IF YOU HAVE A HERCULES MONOCHROME GRAPHICS CARD (or equivalent), you must activate the graphics pages. If you have a Hercules clone or are using the Hercules emulation of an EGA clone, this step may be different.

Type: **hgc full**

You must then load the graphics interrupts.

Type: **int10**

D2. IF YOU HAVE A COLOR GRAPHICS CARD (or equivalent), you must load the extra graphics characters by running the program GRAFTABL.COM supplied on your DOS diskette.

Move to the directory with your DOS system files and

Type: **graftabl**

E. IF YOU HAVE AN 80286 or 80386 COMPUTER AND DO NOT HAVE A MATH COPROCESSOR.

Type: **set no87=0**

You should now continue with the instructions of Section 7.2.3.

Floppy Disk System. You should already have completed the steps in Section 5; if you have not, do so now. The instructions that follow must be completed every time you reboot your computer. Use the DOS AUTOEXEC.BAT file to automate some or all of these steps. If you follow these instructions, your data files will automatically be put on the disk in Drive B:.

A. Put the UnkelScope Working Program disk into Drive A:.

B1. IF YOU HAVE A HERCULES MONOCHROME GRAPHICS CARD (or equivalent), you must activate the graphics pages. Place the disk labelled PC-SUP into Drive B:. If you have a Hercules compatible or are using the Hercules emulation of an EGA card, this step may be different.

Type: **b:\herc\hgc full**

You must then load the graphics interrupts.

Type: **b:\herc\int10**

B2. IF YOU HAVE A COLOR GRAPHICS CARD (or equivalent), you must load the extra graphics characters by running the program GRAFTABL.COM supplied on your DOS diskette.

Insert your DOS diskette into Drive A: and

Type: **graftabl**

C. Put a formatted diskette into Drive B:. (Data will be stored here.)

D. Set the Path to include Drive A:.

Type: **path=a:**

E. IF YOU HAVE AN 80286 or 80386 COMPUTER AND DO NOT HAVE A MATH COPROCESSOR.

Type: **set no87=0**

F. Set Drive B: as the default drive.

Type: **B:**

You should now continue with the instructions of Section 7.2.3.

7.2.3 Running UnkelScope as a Quick Checkout

When you have set up according to Sections 7.2.1 and 7.2.2, type:

scope

The screen will clear and the UnkelScope Welcome Message will be displayed. A typical message (but not necessarily the one you will get) is shown in Figure 7.1. If you do not get any message, read Section 8.

If the message appears, read it to see that the configuration choices are correct. Only the choices that are actually used are indicated in the Welcome Message. If the choices are NOT what you expected, read Section 8.

```
                    Welcome to UnkelScope
     Level 2+ (640k byte) for the IBM DACA Board (Rel 3.05b)
        Copyright 1984 M.I.T., 1985,1987 Unkel Software



                          DOS 3.30
                 First Directory Searched:A:
                Second Directory Searched:A:\

Graphics [IBM CGA & Mono    /Equal]     Printer [IBM Graphics           ]
Computer [IBM PC    or equal ]       Coprocessor [Not Installed ]


        IBM DACA: Adapter 0;    -5 to  5 Volt Input Range
                     IBM DACA Internal Clock
        Digital-to-Analog Output Range -5  to  5 Volts
                4 Channels each with  4096 points


                   Press Spacebar to Continue
```

Figure 7.1 Typical Welcome Message. You should examine the actual message you get to determine that the configuration choices are correct.

If the messages are as you expect, you should make two (2) copies of the message screen. One is to install in the UnkelScope Users Guide so others will know that the system is properly configured; the other is to return with the Software Registration materials. To make a copy of the screen hold down the shift key and press the "PrtSc" key. Do this twice to get two copies.

Once this is completed, press the spacebar to continue. The screen will again clear. The UnkelScope command menu will appear. For this brief test you will sample with the program using the default settings for the sampling parameters.

Press the right arrow key (located on the "numeric" keypad to the right of the keyboard). Then press the enter key.

The screen should clear and display a plot of data. The system will sample the input to the first analog channel at 5 or 20 hertz (50 or 200 milliseconds between samples) and plot values to the screen. The voltage level shown will be that sampled. If you have left the input open, the voltage sampled will be either the maximum or minimum value as set by the range you chose; thus the voltage may be "pegged" to the top or the bottom of the displayed screen. Note that the default setting has 0 volts at the bottom of the plotted region.

If the level is not correct, read Section 8. If it looks correct press the Esc key. Then hold down the Ctrl key and type the character C several times. Respond to the prompt in the upper left hand corner with a y and press enter. This will force an exit from the UnkelScope program.

You are now ready to complete the process by documenting the selections you have made and by completing the Software Registration materials.

7.3 Documentation, Registration and the Final Step

You should now document the configuration of the system. Open the UnkelScope Users Guide to the page that contains Figure 5.2. Cut out one of the configuration screens and tape it into the documentation notebook. Then complete the information in the Figure caption.

Next, find the Software Registration sheet and envelope. To be eligible for software updates and bug fixes with UnkelScope Level 1 or 2+, you must return the registration sheet. Please spend a few minutes to fill it out and insert the registration sheet and the second copy of the UnkelScope Welcome Message into the pre-addressed envelope.

The final step is to use the software. You are now ready to use the UnkelScope Users Guide.

Thank you for selecting UnkelScope and please feel free to contact Unkel Software with questions or comments as they arise.

8. Troubleshooting

This section describes the most likely problems and solutions to problems encountered in setting up and using the software. The errors are organized by the time when they might occur, i.e. messages that will occur before the program has printed anything to the screen come first.

In any case, if you need assistance, call the Unkel Software service number and we will help you. If you are overseas, you may find it more convenient to send us a message by FAX. (See Section 8.5.)

8.1 Errors Encountered Before the Welcome Message is Printed

PROGRAM TOO BIG FOR MEMORY. This indicates that there is not enough memory for the program. One possibility is that you have selected the wrong program diskette. If you have less than 640k bytes you must use the 384k byte version of the program. If you have only the minimum required memory, you may find that the program is a tight fit and you may not be able to load additional programs and still run UnkelScope. For example, you will not be able to have a print buffer and may not be able to load many "resident" DOS functions. If you are unsure of how much memory you have you can use the DOS function CHKDSK, which gives a listing of the memory available. If CHKDSK indicates you have the memory you expected, call the Unkel Software service line. Another possibility is that a section of the disk was incorrectly copied. Try the original copy to see if it gets the same error but keep in mind that the write protect tab in the original disk will prevent it from running the program.

BAD COMMAND. You have forgotten or incorrectly used the path command, have chosen the wrong diskette, or did not correctly complete the installation procedure. List the files in the \uscope directory or on the disk to see if it is correct. Section 5.2.1 or 5.2.2 gives a list of the files that must be present to run the software.

FILE NAME MISSING OR BLANK.PLEASE ENTER NAME UNIT 4? The diskette or disk subdirectory does not contain all the necessary files. List the files on the disk with the DIR command; compare the files with the list in Section 5.2.1 or 5.2.2. If all the files are not on your disk, you should redo the configuration setup step. If running from floppy disks you may have invoked the program on Drive B: when DOS was pointing to Drive A:, or vice versa. For a floppy disk system you must:

a) have the Program Diskette in Drive A:;
b) have a formatted floppy diskette in Drive B:;
c) have entered the DOS Path command **path=a:\;**
d) selected Drive B: as the default drive by typing **b:**; and,
e) started UnkelScope by typing **scope** to the B> prompt.

Not performing any one of these steps could be your problem.

For data acquisition systems consisting of a remote device connected through the Serial Line (RS-232) adapter of the computer:

ERROR IN DEVICE. While UnkelScope is setting up the Welcome Message it may reset the remote device and request the remote device to return the Welcome Message. If these steps are not comleted properly, an error message will be printed to the screen in reverse video and the bell will sound three times. If this happens the probable causes are:

a) the communication link is faulty (eg. the cable is not plugged in);
b) the remote device is not turned on; or,
c) the remote device or the Serial Line Board has been damaged in some way.

If this happens, you should exit UnkelScope, turn off the remote device and check the communication cable. Then repower the remote device and try UnkelScope again.

8.2 Errors Encountered After the Welcome Message is Printed.

PROGRAM DOES NOT RESPOND TO THE ARROW KEYS. The "numeric" keypad does double duty. The "NumLock" key located to the upper right toggles the numeric keypad between the direction keys and the numbers. Try pressing the NumLock key and releasing it. If this doesn't resolve the problem, try restarting the program. If the problem still persists, call the Unkel Software service line.

NO REVERSE VIDEO IN THE SETUP OR GRAPHIC EDITING SCREEN. You may have selected the wrong graphics support or you did not perform the special setup procedure required by your graphics system. See Section 4.3.

STRANGE NUMBERS FOR SCALE LIMITS AND/OR NO PLOT. If you have an 80286 or 80386 computer with no math coprocessor, you may have forgotten to type the command "set no87=0". This must be typed each time you reboot the system or be included in the AUTOEXEC.BAT file.

8.3 Errors Encountered While Sampling

NO PLOT BUT SCAN ENDS. You may have selected the wrong graphics support, i.e. you selected the Hercules Graphics Card when you have a Color Graphics Adapter, or you did not perform the special setup procedure required by your graphics system. See Section 4.3.

PROGRAM DOES NOT PLOT POINTS WHEN USING THE DEFAULT SETTINGS. In this case the most likely cause is that the Base Address is not consistently selected, or that there is a conflict between the Data Acquisition Board and another board in the computer. If this seems to be the problem, either remove the potentially conflicting board or alter the Base Address of the Data Acquisition Board. (Remember that the Base Address must be set to the same value in the Configuration program as it is set by the hardware on the board itself.)

VOLTAGE VALUE SHOWN DOES NOT AGREE with the value measured by some other device. Most likely you have not set the hardware switches and chosen the software configuration in a consistent fashion. Double check that the configuration chosen agrees with the selections on the board. Another possibility is that the connections to the board are not properly done. Check that you have properly installed any cables and connected your voltage source to the proper lines.

COMPLETELY BLANK SCREEN FOR SAMPLE/DISPLAY. You may have selected the wrong graphics support. Check with Section 4.3.

LABELS AND GRAPH BOUNDS BUT NO PLOTTED POINTS IN SAMPLE/DISPLAY. A symptom of incorrect graphics support. Check with Section 4.3.

8.4 Errors Encountered While Printing

UNUSUAL CHARACTERS WHERE BOXES SHOULD BE. Your printer needs to be set up for the IBM special graphics character set. This usually requires setting a switch on the printer. See your printer manual for switch settings.

8.5 Phone Service

In any case, if you have made your best attempt to get the equipment running but it still does not function, call us.

If you are installing your hardware together with the UnkelScope software and you experience difficulties not adequately discussed in this document, you should call the Unkel Software Phone Service number listed below. It is often difficult for the user to determine whether a particular problem is caused by hardware or software. At Unkel Software we can assist you with software problems and can help identify hardware problems that require the assistance of the hardware manufacturer or representative.

Unkel Software Phone Service : (617) 861-0181

Unkel Software FAX Line : (617) 861-1850