

GRAPHICS FOR REAL-TIME SIGNAL PROCESSING SYSTEMS

by

M.J. McCann
SACLANT ASW Research Centre
La Spezia, Italy

ABSTRACT A graphics software package able to produce hard-copy plots of multi-channel (of the order of 100 channels) data for real-time applications is described. The system is implemented at present using a lineprinter with graphics capability, and is able to plot multiple curves along or across the paper with full annotation. In addition, intensity modulated plots may be made for use in applications such as satellite imaging. Features of the system are multi-channel capability, small core requirements and lack of restriction on plot length. Also described is a control technique for graphics programs whereby plots may be easily modified and plotting mode changed, allowing a high degree of flexibility as is required during on-line experimental work at sea.

INTRODUCTION

This paper deals with the WARP display subsystem developed at SACLANTCEN for use primarily with the lineprinters used by the Real-Time Systems Department. The principles involved, however, are applicable to many "raster-type" plotting devices; that is, devices which generate a plot from a series of dots drawn on parallel lines. The system used to control the plot output is also described.

The system comprises a subroutine package, which is of general utility outside the system as well as within it, and a control program. The subroutine package is described first.

1. THE PLOTTING SUBROUTINE PACKAGE

The package was developed largely in response to the immediate hard-copy plot requirements of a number of projects in underwater acoustics and oceanography. A list of requirements was drawn up as follows:

- a. A multi-channel plotting facility able to handle up to 100 data channels plotted side-by-side in real time, or as close to it as possible.
- b. An option to suppress hidden lines where curves overlapped.
- c. An ability to plot large numbers (several hundred) of signals from successive events, with the option of plotting "filled wiggles" — that is, curves with the area beneath them filled in.
- d. A variable density plot able to reproduce photographic images (particularly those from satellites) besides data plots.
- e. A fast contouring capability.
- f. Full annotation — axes, grids, text and so on.

These were the firm requirements — in addition it was felt that there should be a facility for plots of two independent variables although almost all the data we handle have only one monotonic independent variable such as time or frequency. This, in fact, influenced the final form of the plotting package very greatly.

We examined all the existing graphics packages and systems we could find, but none could meet our requirements. It was decided that a system would have to be produced in-house at SACLANTCEN, which would ultimately be able to be run as a stand-alone interactive graphics facility or added to other systems to provide a flexible graphics output requiring a minimum of programming effort to use.

The hardware in use at the time already included three Printronix lineprinters. These machines have a graphics mode — that is, they may be programmed to print single dots sufficiently close together to form a continuous line horizontally, vertically or diagonally. They are relatively inexpensive (about \$5,000), use cheap paper, and have proved reliable in use. As one of them is normally included in the hardware of a system as a lineprinter, we decided to develop the package using these machines.

The basic problem in plotting with this type of device is that a line across the paper is only printed once — it is not possible to return and add additional information. Thus, each line must

be absolutely complete, with dots for curves, axes and all text, before being sent to the lineprinter.

The most straight forward way to produce a plot, therefore, is to generate the entire plot as a bit map in an area of core (or some outside storage device), which is then scanned and sent to the printer line-by-line [Fig. 1]. Indeed, for plots with two independent variables this is often the only practical method. The problem is that either a large amount of storage is required or else the plot must be divided into fixed-size pages, presenting a variety of problems.

The alternative to the core image approach is to generate successive single lines of output which are sent to the printer as soon as each is complete. This technique is only suitable for plots in which one variable is monotonic and preferably defined at equally spaced intervals — conditions which apply to at least 90% of experimental data at SACLANTCEN. The advantages are that the plot package as such requires very little storage and there are no inherent limitations on plot length or problems with paging [Fig. 2]. It was decided to make this the basis of the plot package.

2. IMPLEMENTATION

A plotting program using this package normally operates as a loop [Fig. 3], using the number of the printer line currently being assembled as the controlling variable, line 1 being the start of the plot. For each line, the line buffer is first cleared, next a bit is set at the intersection points of each data curve and the current line, then bits for the axes and any characters are set in addition, and finally the line is output. Each routine called has as parameters the current line number and variables specifying at which line its output is to start and stop, together with the parameters peculiar to itself.

The system has four modes of operation. The first is a core image system similar to that shown in Fig. 1. This is primarily used for the output of blocks of text or numbers — in fact, at the time of writing it has been used for nothing else, except test patterns. The core image output routine has been written to allow rotation and enlargement of the core image on output — thus, only a relatively simple routine is required to generate an image in store of alphanumeric output blocks of minimum size (5 x 7 bits per character) and one orientation. The core image area may be of any size, and a routine has been written to generate a vector between any two points on the image. The image output routine is compatible with the rest of the system so that text may be added to any plot.

Secondly, curves may be plotted across the paper, one beneath the other. Each line number is interpreted as a data value relative to the current origin, and the data series scanned to determine whether this level has been crossed by the curve; if so a bit is set in the line buffer. For filled wiggle displays [Fig. 4] bits are set as long as the data level is greater than or equal to the line level. This mode is particularly suited to this type of display — for curves which overlap a "cylindrical buffer" of as many complete data curves as can overlap must be maintained, each curve being plotted on each line [Fig. 5]. Hidden lines may be suppressed by, in effect, a "reverse filled wiggle" clearing bits as long as the curve being plotted is above the current line level.

Note that in order to produce a continuous line it is necessary to set bits as long as the curve is in the range $V \pm I/2$ where

V = current line value

I = Value interval between successive lines [see Fig. 6]

The limitations of this mode are:

a. The number of points plotted on one curve is limited by the line length — 792 for the Printronix, though normally limited to 700 to allow room for annotation.

b. The degree of overlap of curves allowable is limited by the number of data sets which may be held in store at one time.

c. Large-amplitude plots are slow to produce as more lines are required for each curve.

The third plotting mode uses the printer to plot along the paper in a manner similar to a pen or galvanometer recorder. Here, the line number is proportional to the independent variable, and the data value determines the position of the dot along the line. The origin of each curve is normally offset by a constant amount, but the offset may be zero or varied to give grouping of curves if required.

This type of plot is very suitable for multiplexed data, as each line requires one sample from each curve. This is a problem for non-multiplexed data, as the curves must be in effect, re-multiplexed before display. In the Hewlett Packard system this may be conveniently achieved by reading all the curves into a two-dimensional array in core, using the extended memory access (EMA) facility, and reading the data

out line by line. EMA is normally required as an array typically occupies around 50K words in our applications. Filled wiggles and hidden lines, as well as simple curves, may be output in this mode; hidden lines are particularly simple to implement by plotting only points to the right of the last plotted point on each line. To obtain continuous curves in this mode it is necessary to retain the positions of the last plotted points for each curve on the previous line, and to print all points between these positions and those for the current line [see Figs. 7 and 8].

Finally there is a grey scale plot mode. The effect is achieved by using pixels of 3 x 3 dots with from 0 to 9 dots printed. This gives 10 grey levels and a plot width of up to 264 points. When designing the pixels care was taken to ensure that areas of similar pixels showed distinctive dot patterns when viewed close-up as an aid to quantitative interpretation of the plot [Figs. 9 and 10]. This plot mode is the only hard copy grey scale output available to us except for photographs of CRT monitors. One problem is that the plot is distorted because the pixels are not square, due to the difference in horizontal and vertical dot spacing (72 dots per inch vs 60 dots per inch). This has not, so far, proved a serious embarrassment; if it were, a 6 x 5 pixel would be square and give more grey shades but less resolution. The grey scale plot also gives a useful contouring effect in some applications, and has the advantage of providing a qualitative insight into the shape of the surface as well as a quantitative contouring [Fig. 11].

Annotation of plots with X and Y axes, grids and blocks of text is possible. Axes are plotted as, in effect, special curves, being generated line by line. Text and numerical output, however, is written into buffers as bit patterns, and output as though each pattern were an X-Y plot, as already described. The axis generating routine may also be used to generate X-Y grids.

There are a number of further routines for special displays. For instance there is a routine which will output a core image stored as a series of dot positions, which is more economical on storage than a bit map for "sparse" images such as coast-lines. Another routine will plot an annotated grey scale in standard format. A third will plot lines between corresponding points on two separate curves, giving a band of varying width.

3. CONTROL

The control system is similar to that used for the rest of the WARP I system. The information required to specify a plot is held in ASCII format, stored as a file in the standard HP file manager system. This type of file may be read and edited by the operator, using the normal file manager and editing facilities, and also read by the program line by line.

A typical display file listing is shown in Fig. 12. Each line starts with a two-letter code, which is followed by the parameter or parameters to be read. Anything following the parameters is ignored by the program; thus this area is used for descriptive comments. A special code, two asterisks, is used if a whole line of comment is to be included.

When operating, the program runs in a loop which reads each successive line, identifies the parameter, reads it and continues to the next line. The loop is terminated by the /E code at the end of the file. Thus the parameters may be in any order, or repeated, without causing any problems.

It was found very useful to write a special editing program for the plot files, so that the operator could, for instance, set all the channel maxima to a given value at once, rather than having to edit line-by-line. This program enables a file to be generated or updated by simply typing the parameter code followed by the value — the comments are added automatically when the file is re-written.

The plot program always reads the same file when starting a plot. The file manager is used to copy the relevant file into this standard file before the program is run; in fact, a so-called "transfer file" is used to load all the control files for the WARP I system and run the first program. Files may thus be written and stored in advance of the program run, with descriptive names to assist the operator.

This control system has been found very convenient in use — the average time taken to modify a plot file during a recent trial was around ten minutes. It was also found useful to generate "blank" file printouts — that is, printouts in the standard format, but without the numbers, which could be filled in and then quickly copied by the operator using the special editor already described.

4. FUTURE DEVELOPMENTS

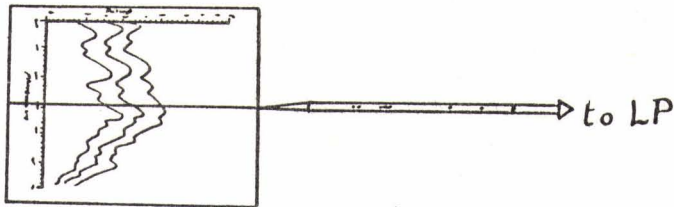
As may be seen in Fig. 12, the plot control file is currently used to specify which data are to be plotted (via the number of channels displayed (ND) and the identifying column (CN) in the channel specification lines). This function will be separated to increase the generality of the plotting program.

We have found that the lineprinters give good quality output on inexpensive paper, and these will become our standard plotting device. Work will be done on increasing the efficiency of the subroutine package by using assembler and microcoding parts of it — most of the work carried out by the routines consists of logical operations such as bit packing and unpacking, "AND" functions to add bits to lines, shifts and so on which are not handled efficiently in FORTRAN but are easily implemented in assembler and microcode.

The system will shortly be extended to make use of the Lexidata colour display and Applicon colour plotter. We may also add an electrostatic printer/plotter to give better resolution and speed.

CONCLUSIONS

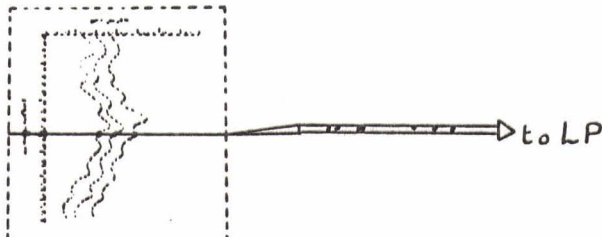
This plot package for multichannel applications is working satisfactorily and provides an efficient and flexible plot facility. The writing did not require a great deal of resources — about two man-months. The principle of control of systems via ASCII files has proved very successful and will be further refined and extended during the coming year.

"Core Image" System

The image or image page is completely specified in core as a bit map, then scanned and output line by line.

- Only practical method for plots of two independent variables
- Require large core area, or a complex paging system

FIG. 1 CORE IMAGE SYSTEM

Alternative System (as used in WARP)

Only one line of the plot at a time is imaged in core, and output when complete.

- Suitable for plots with one independent monotonic variable.
- No paging of plot or limitation of plot length.
- Minimises core requirements.

FIG. 2 ALTERNATIVE SYSTEM (AS USED IN WARP)

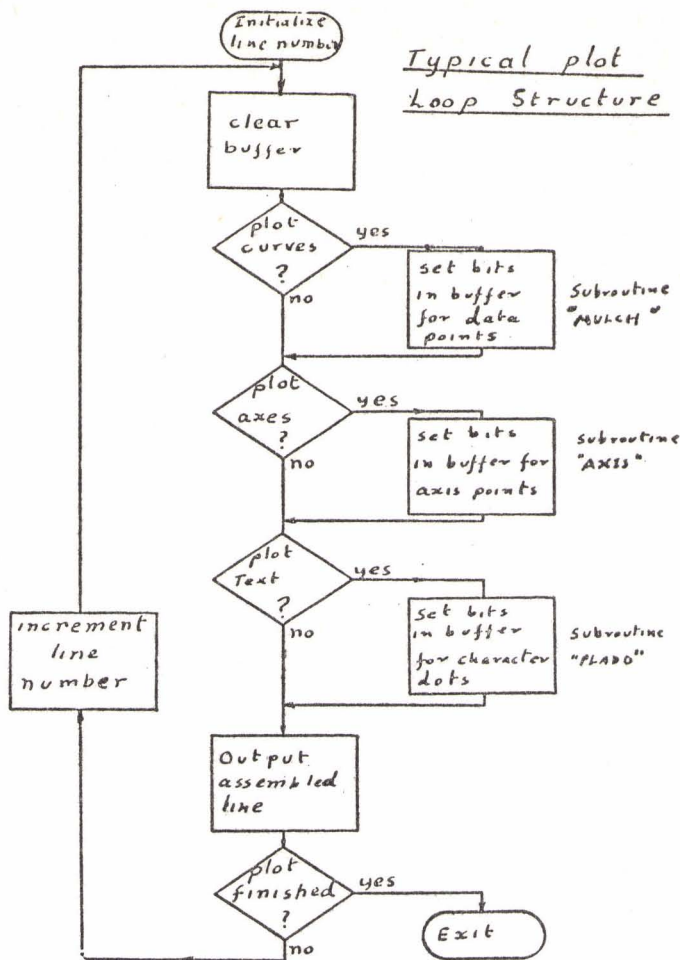
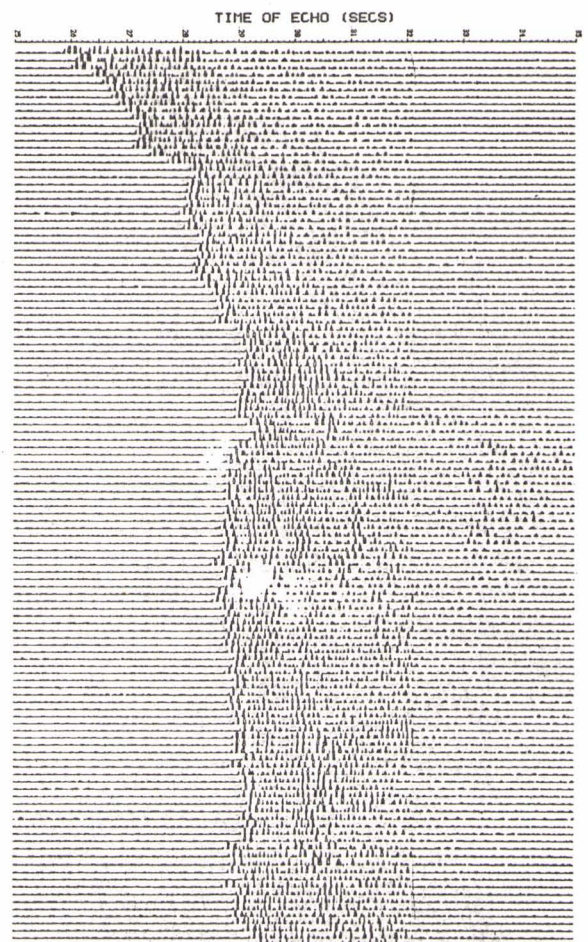
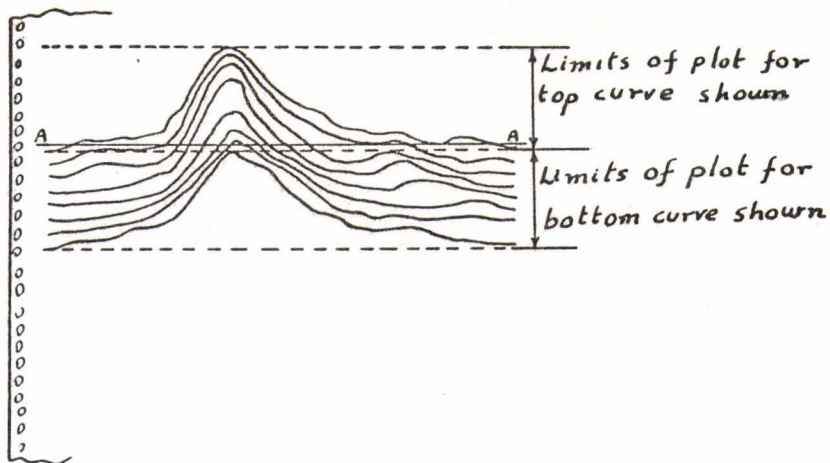


FIG. 3

TYPICAL PLOT LOOP STRUCTURE

FIG. 4

TYPICAL "FILLED WIGGLE" DISPLAY FROM
SUB-BOTTOM PROFILER



Here, the curve separation = $\frac{1}{7}$ of the plot height. Thus, line A-A of the plot can intersect 7 data curves, requiring seven curves to be in store and examined for each line plotted.

FIG. 5 PRINCIPLE OF ROTATING BUFFER STORAGE WHEN PLOTTING OVERLAPPING CURVES ACROSS PAPER

Each line corresponds to a data value :

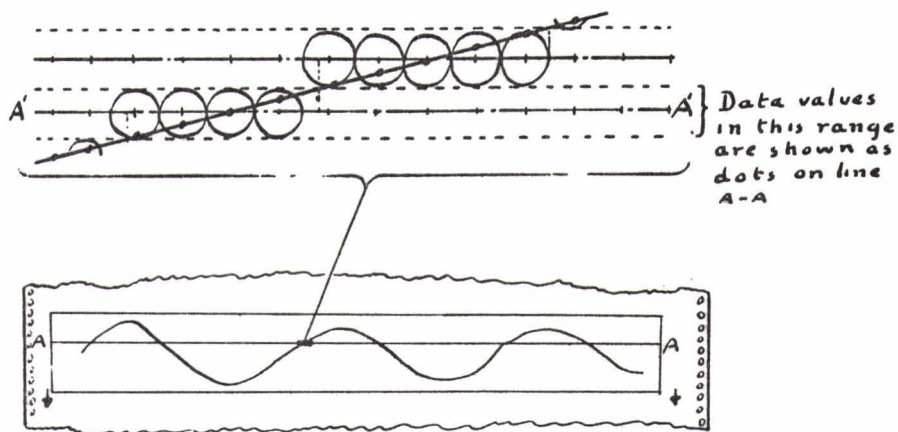
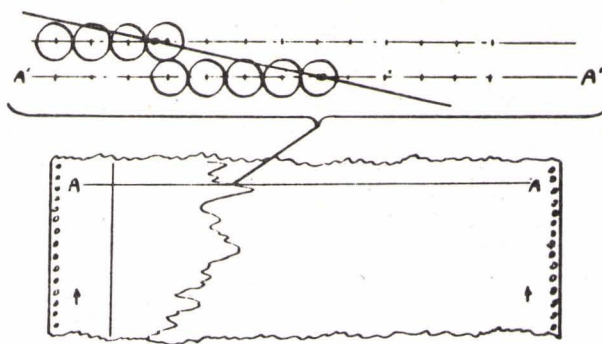


FIG. 6 CURVE PLOTTING ACROSS PAPER



Here, each line corresponds to a data point, and the dot position along the line to the plotted value.

FIG. 7 CURVE PLOTTING ALONG PAPER

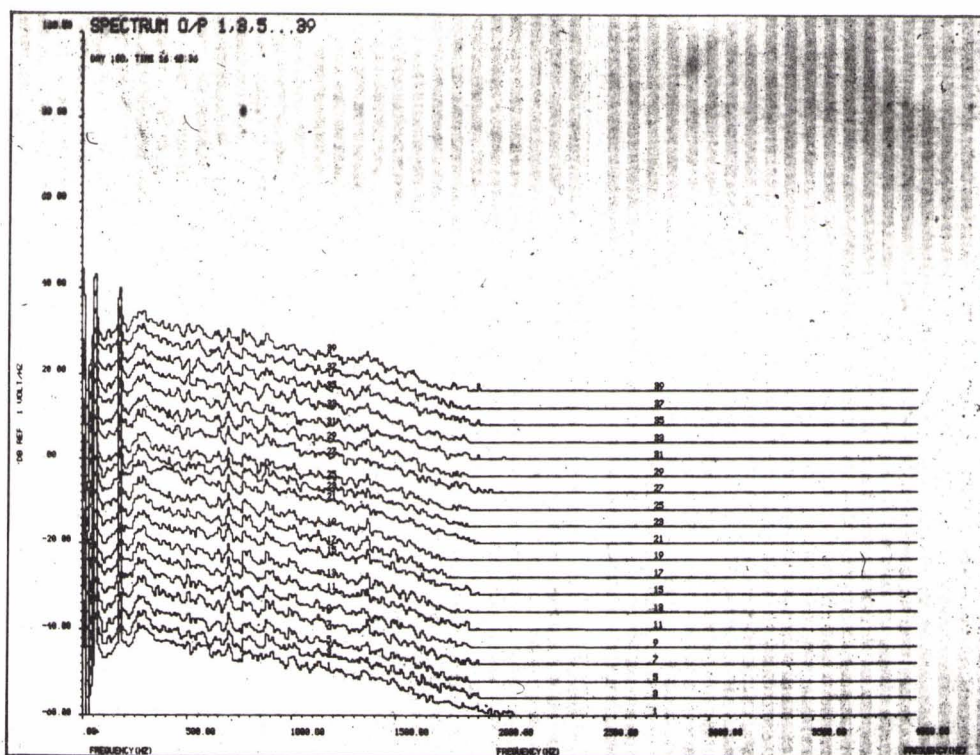


FIG. 8 TYPICAL PLOT OF SPECTRA, PLOTTING ALONG PAPER

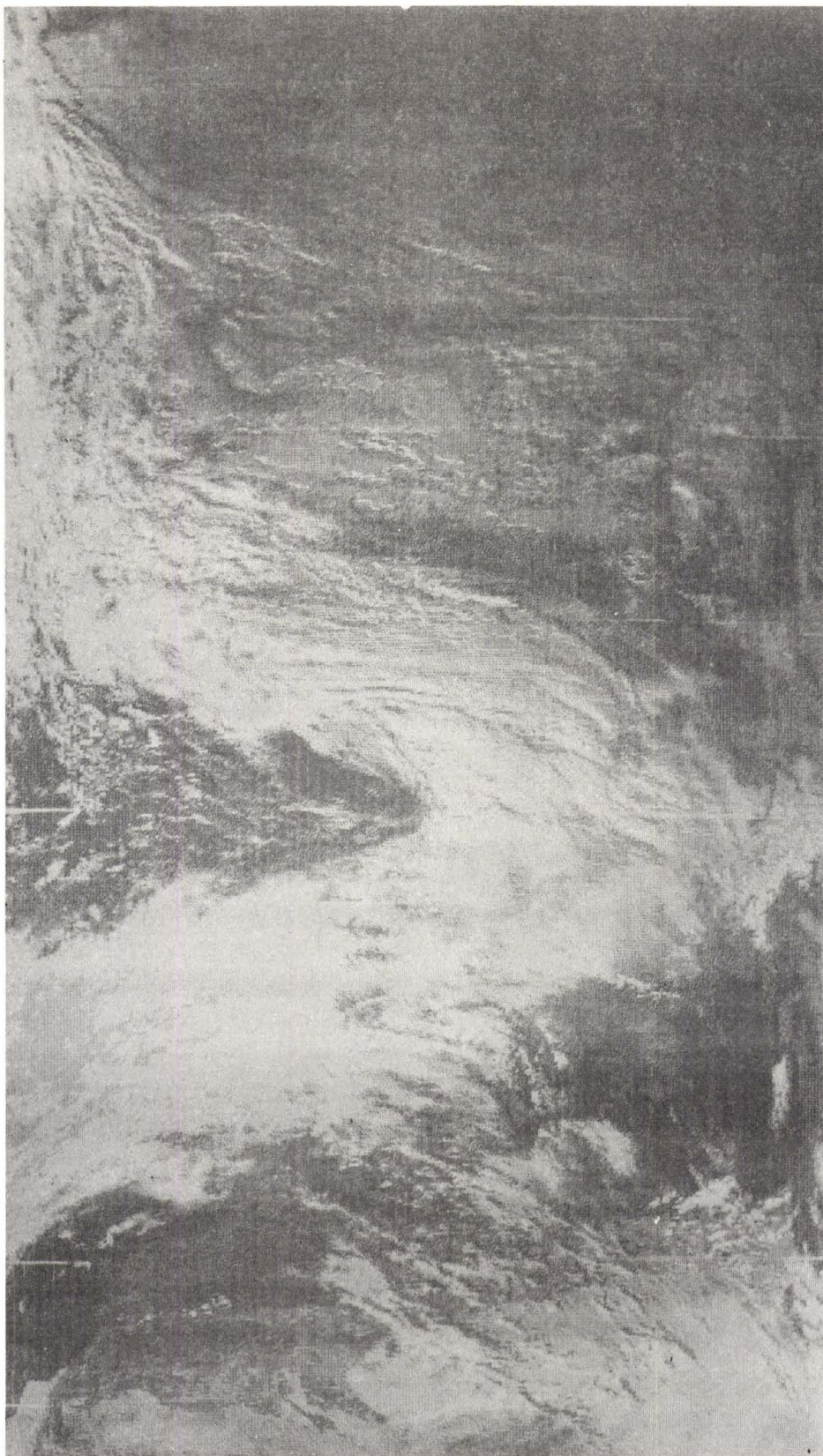


FIG. 9 TIROS-N VISIBLE CHANNEL - N. AFRICA, EUROPE AND E. ATLANTIC
OBSCURED BY CLOUD.

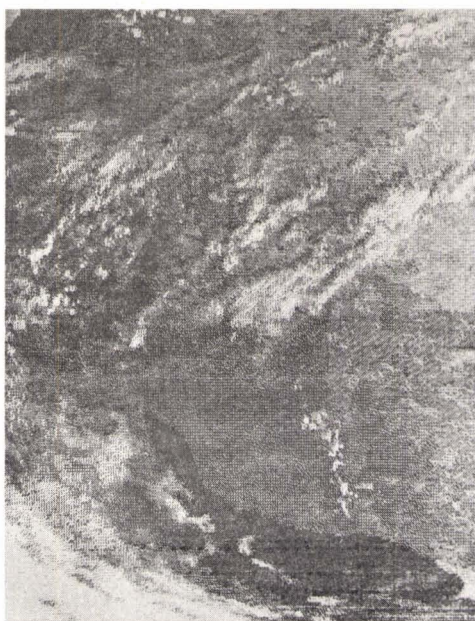
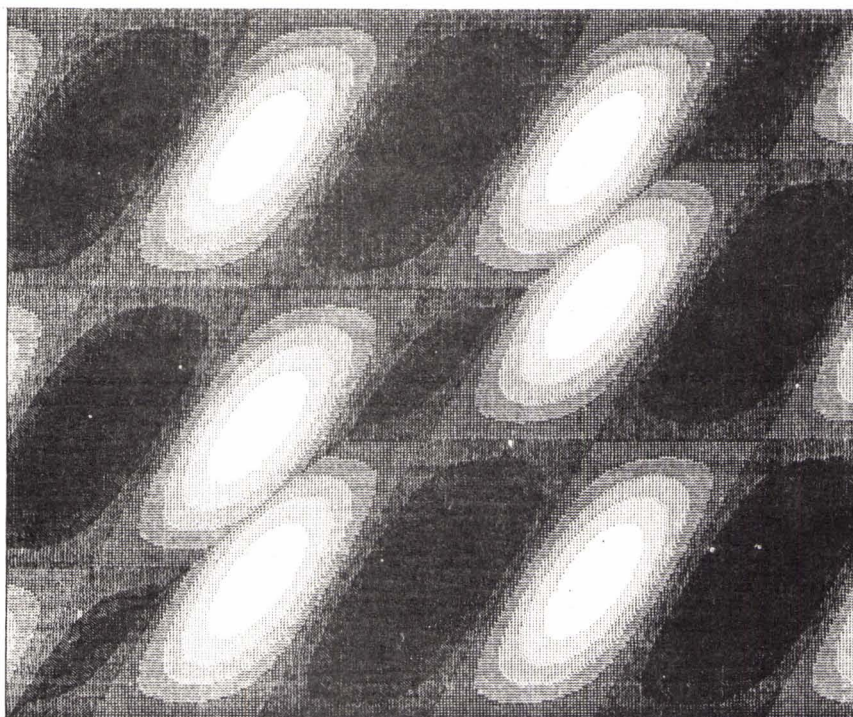


FIG. 10 PORTION OF TIROS-N PHOTOGRAPH DISPLAY (ACTUAL SIZE) SHOWING CLOUDS OVER N.W. AFRICA. ONLY 1 IN 8 POINTS ARE PLOTTED IN THIS CASE



GREY SCALE TEST PLOT - MOVING SIN/COS SUM



FIG. 11 EXAMPLE OF GREY SCALE TEST PLOT, SHOWING INHERENT CONTOURING.

JE T=00004 IS ON CR32767 USING 00005 BLKS R=0004

```

WP 600      *PLOT WIDTH (700 MAX.)
WI 300      *CHANNEL WIDTH
PL 1        *PLOT TYPE : 1 FREQUENCY, 2 TIME SERIES
MP 3        *PLOT MODE: 1: SIMPLE, 2: FILLED WIGGLE, 3: HIDDEN LINE
CF 1        *CONDENSATION FACTOR
CM 1        *CONDENSATION MODE
**          (0: 1 IN N, 1: AVERAGE, -1: LARGEST ABS.)
ND 4        *NUMBER OF CHANNELS DISPLAYED
XL .0       *LOWER X-AXIS VALUE
XI 100.0    *X-AXIS INCREMENT
XU 4000.0   *UPPER X-AXIS VALUE
YL .0       *LOWER Y-AXIS VALUE
YU 2.0      *UPPER Y-AXIS VALUE
NY 11       *NUMBER OF Y-AXIS NUMBERS
TX FREQUENCY (HZ) @@
TY VOLTSE   @@
TP TEST OUTPUT@ @@
**
** CH <N> TO CHANGE SPECIFIC CHANNEL PARAMETERS.
** MA TO SPECIFY ALL MAXIMA, MI ALL MINIMA,
** SP TO TYPE ALL POSITIONS (IN AXIS UNITS)
** SK TO SPACE A SET OF CURVES BY A CONSTANT AMOUNT
** CN TO TYPE DATA CHANNELS TO BE DISPLAYED
** CI TO TYPE CHANNEL IDENTIFYING NUMBERS TO BE PLOTTED.
** CC TO TYPE CHANNEL CORRECTION SPECIFIERS.
**
**          MA          MI          SP          CN          CI          CC
CH 1        1.000      .000      .000      1      11      0
CH 2        1.000      .000      .100      2      22      0
CH 3        1.000      .000      .200      3      33      0
CH 4        1.000      .000      .300      4      44      0
/E

```

FIG. 12 EXAMPLE OF PLOT CONTROL FILE BASED ON 2-LETTER CODE WITH COMMENTS.