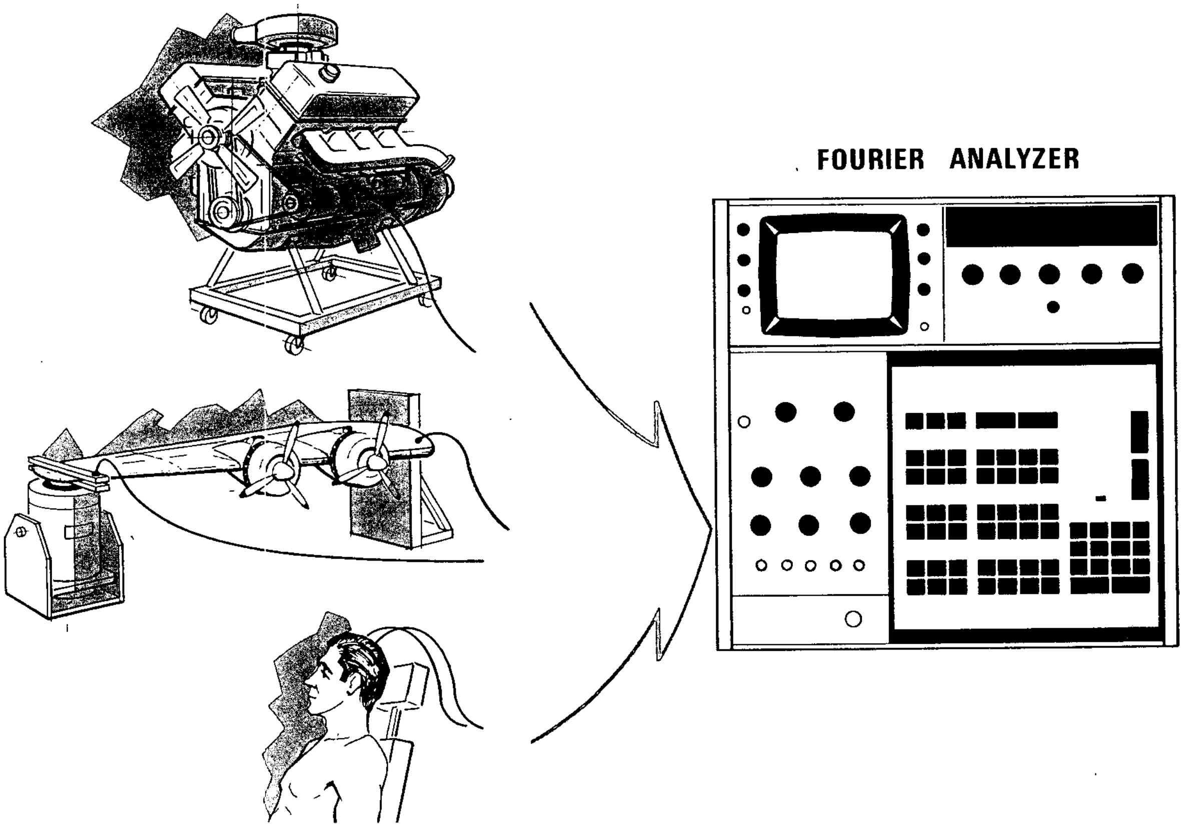
APPLICATION NOTE 140-0

# Fourier Analyzer Training Manual

## Summary of this Manual

This manual first describes some typical applications of the Fourier Analyzer, such as analysis of: ship sounds, vibrations in engines and wing structures, brain waves, echoes from underground test explosions in oil research, etc. Then it shows how data is processed by pressing keys on a keyboard, and how individual mathematical operations can be linked together from the keyboard, to form programs. A power spectrum program is described, for detecting signals buried in noise. The transfer function and coherence function used in systems analysis are explained. The last section of the manual is an extensive discussion of the mathematical background of the Fourier Analyzer.



HEWLETT hp PACKARD

# TABLE OF CONTENTS

Page

SECTION I – WHAT IS A FOURIER ANALYZER?

SECTION II – HOW FOURIER ANALYZER WORKS

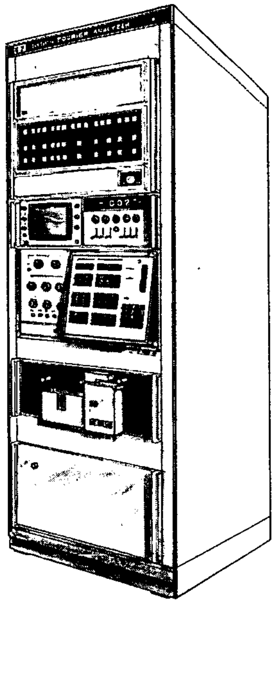
SECTION III – SOME FINE POINTS

SECTION IV – FOURIER TRANSFORM: MATHEMATICAL BACKGROUND

# SECTION I

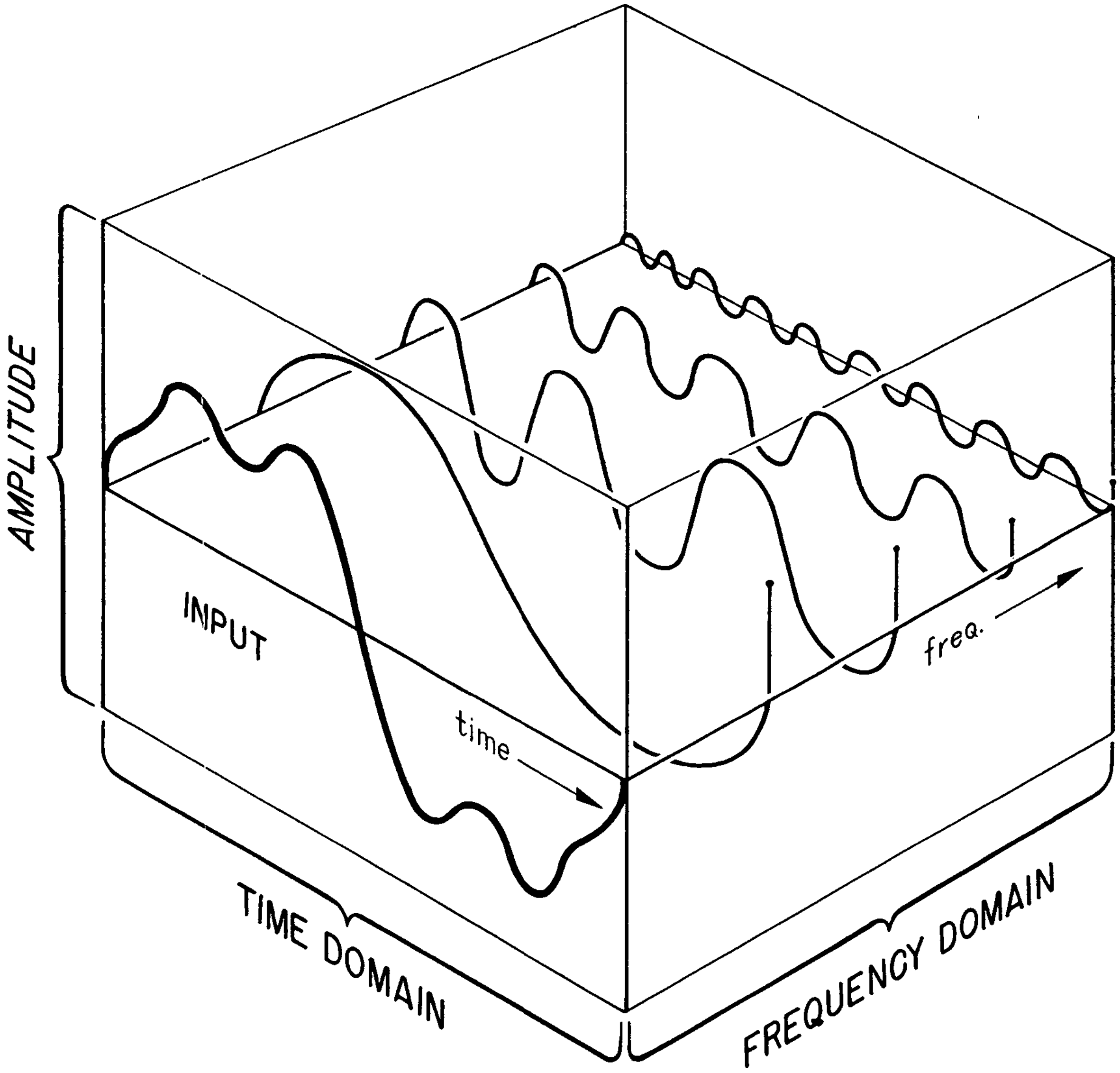
## What is a Fourier Analyzer ?

## Typical Applications



## WHAT IS THE FOURIER ANALYZER?

An instrument that can break down any input that varies with time, and show its component frequencies. It does this digitally, which means it is more accurate and more flexible than analog machines such as the spectrum analyzer and wave analyzer. It is faster than wave analyzers and faster than most spectrum analyzers. Main feature is a Keyboard on which the user can punch keys for a variety of mathematical functions to be performed on the frequency data. No knowledge of programming is required.



*Signal Analysis – Given the Input, to Find the Frequency Components.*

## WHAT ARE SOME TYPICAL APPLICATIONS?

### Sonar

Submarine listens to sounds in water, uses Fourier Analyzer to calculate power spectrum of sounds, showing their component frequencies and amplitudes. Then compares this measured spectrum to a set of standards or ideals for particular objects, determines if object heard is surface vessel, school of fish, whale, or other submarine.

*Underwater Acoustics – Analyze Underwater Sound, Determine if Object is Sub, Whale, etc. If Sub, whether Friend or Enemy.*

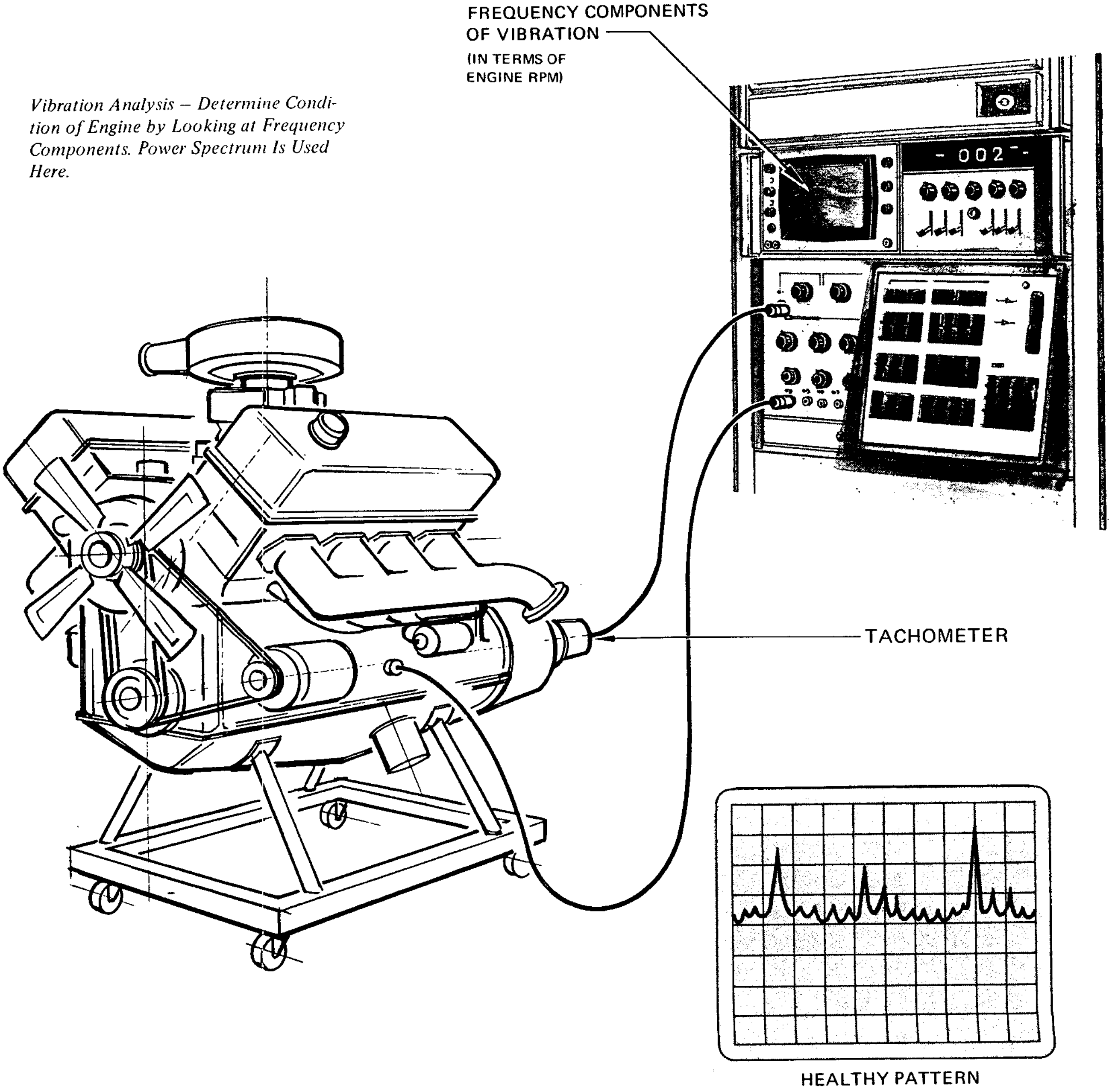
## Systems Analysis

### POWER SPECTRUM

Put transducer on automobile engine, use power spectrum to determine frequencies present; compare against known healthy pattern of frequencies. The power spectrum can be measured in orders of engine RPM for a spectrum that is stable even if the speed changes.

### TRANSFER FUNCTION

Or put transducers at two points in a system (typically input and output) and determine the transfer function between the two points. That is, determine how output is related to input. For example, test model of airplane wing using shake table vibrations as input, wing vibrations at certain point as output.



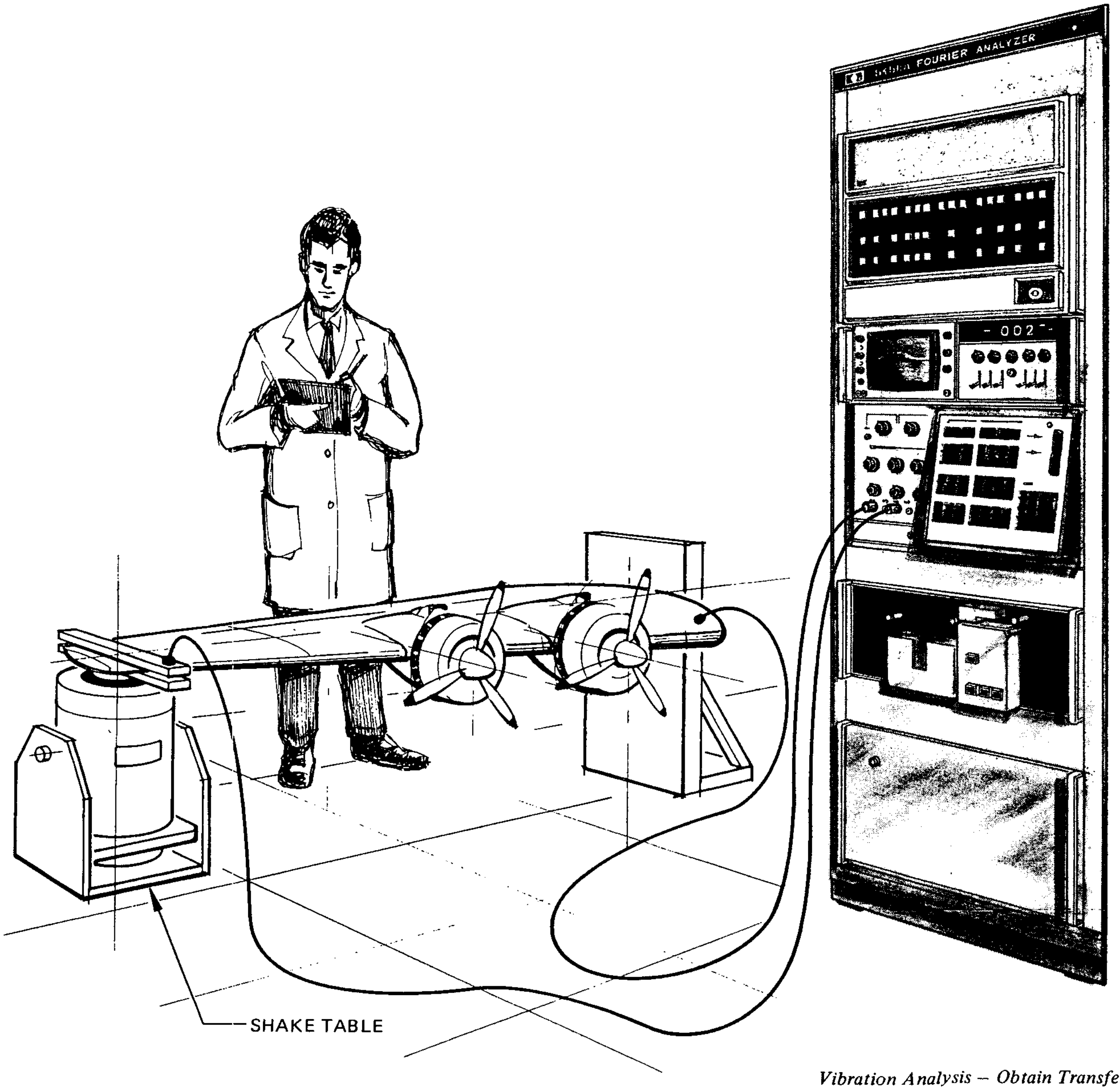
*Vibration Analysis – Determine Condition of Engine by Looking at Frequency Components. Power Spectrum Is Used Here.*

## COHERENCE FUNCTION

To test validity of transfer function over the frequency range, calculate coherence function. This function gives relative indication of causality between output and input. Low value at any frequency indicates transfer function may not be accurate at that frequency.

## OTHER APPLICATIONS

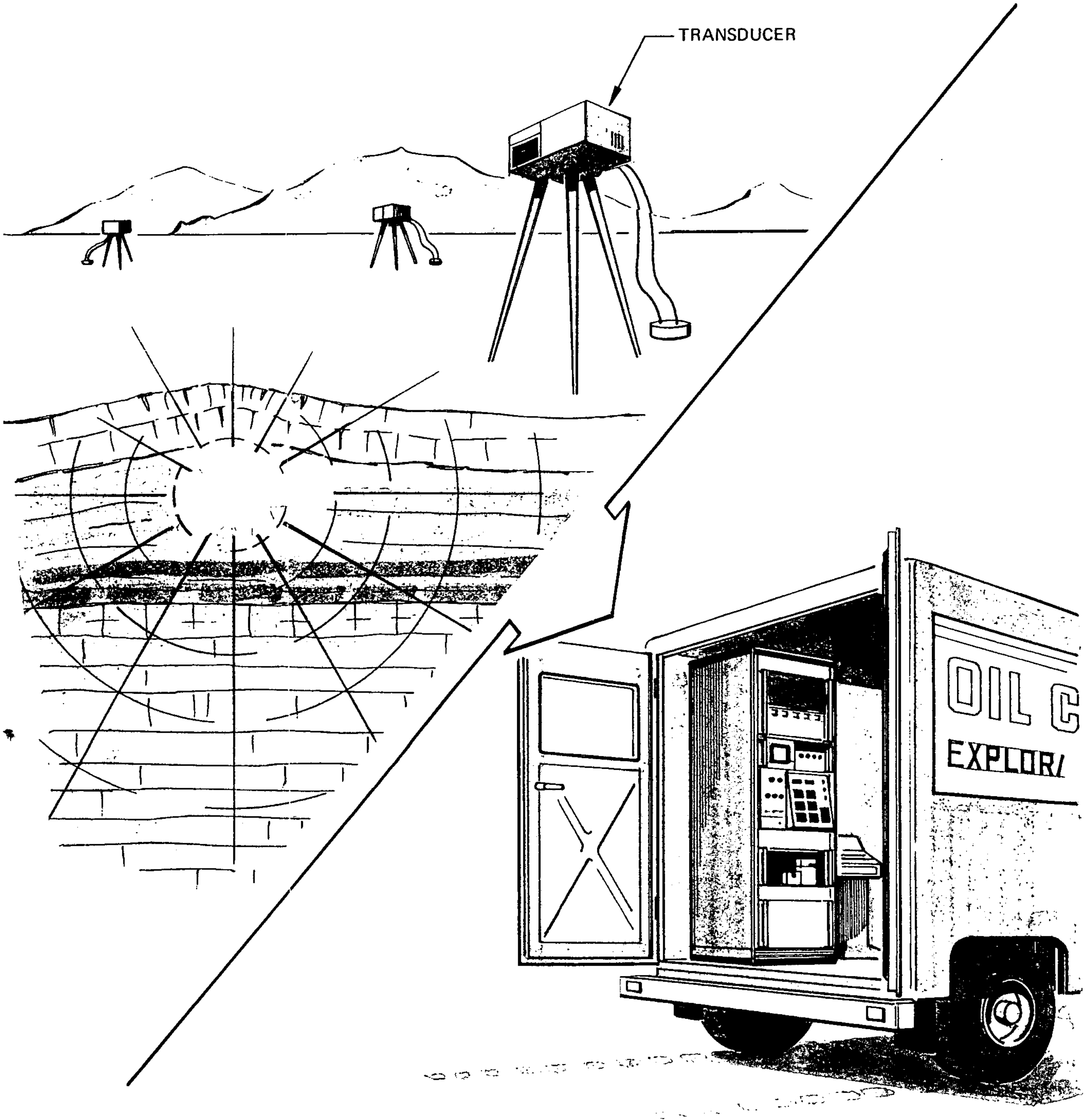
Other typical system analysis applications: determining transfer function between front wheels and steering wheel of car; determining filter response; determining transfer function of body organs, and many others.



*Vibration Analysis – Obtain Transfer Function of Model of Airplane Wing*

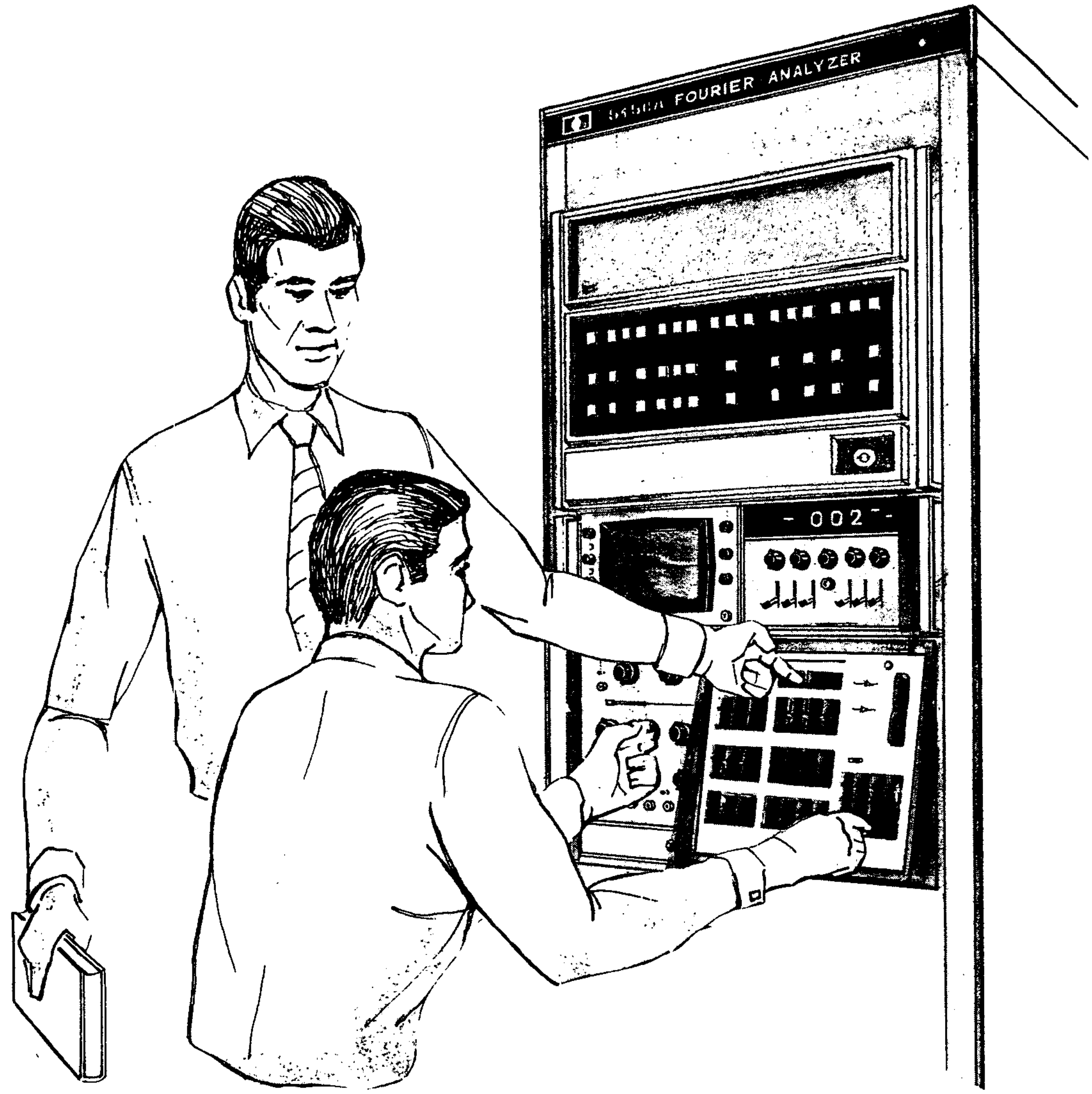
## Oil Research

Fire small explosion on surface of the earth, have shock-sensitive transducers in ring around the explosion, record resulting vibrations in transducers, analyze in Fourier Analyzer to determine kind of echoes present. Certain echoes indicate oil.



## Educational

Use instrument to demonstrate math functions such as Fourier transform, power spectrum, cross power spectrum, correlation, convolution, transfer function, coherence function, characteristic function, various weighting functions such as Hanning. All math operations available at Keyboard; easy to go from one to the other, show effect of changing variables, etc.



## WHAT HP INSTRUMENTS IS THE FOURIER ANALYZER SIMILAR TO?

### Wave Analyzer and Spectrum Analyzer

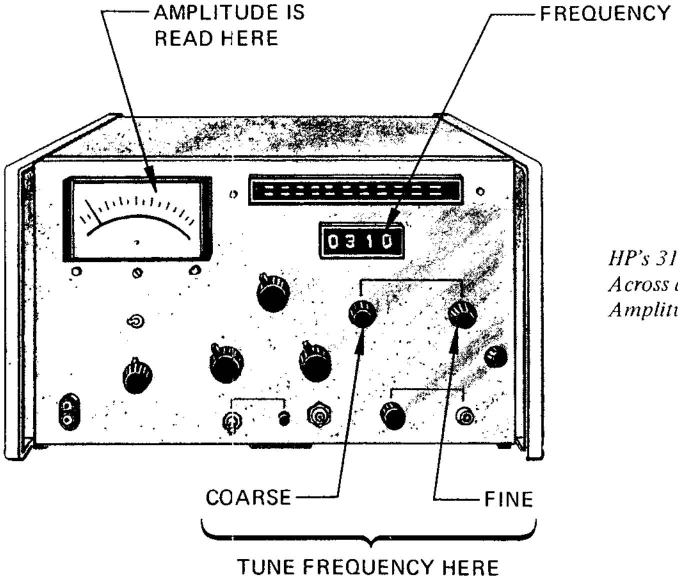
In a wave analyzer, we take a time input, then manually sweep a filter across it to determine the frequencies present. (Large amplitude equals large amount of energy present at that frequency.)

In a spectrum analyzer, the frequencies for a given scanwidth are displayed simultaneously on a scope screen. Here, the filter is swept electronically across the input.

Both these instruments are analog devices.

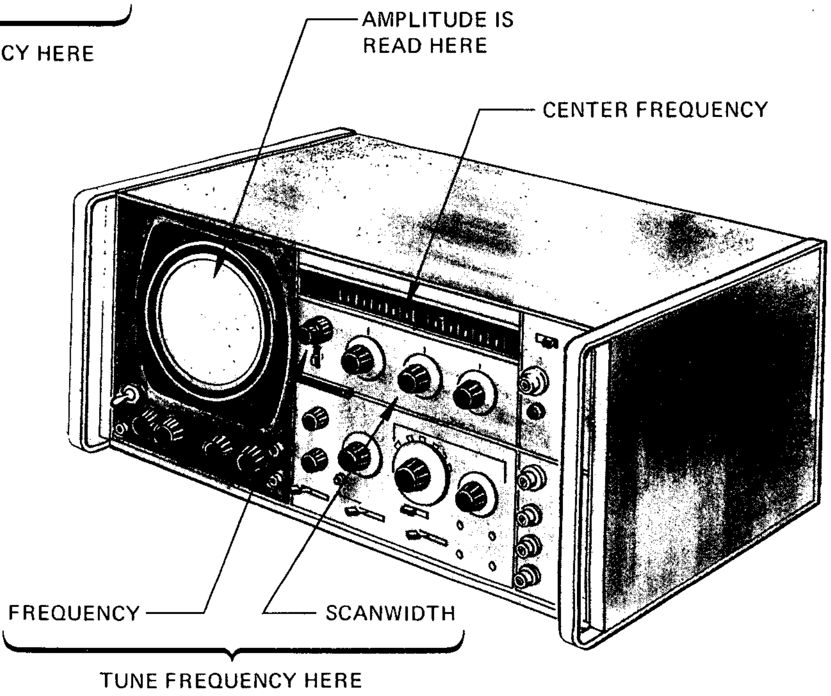
What is the difference between the Fourier Analyzer and these?

It is digital, not analog. This means that it is much faster than a wave analyzer or spectrum analyzer over its frequency range of dc to 25 kHz.



*HP's 310A Wave Analyzer – Tune Across a Frequency Band, Note the Amplitude at Each Frequency.*

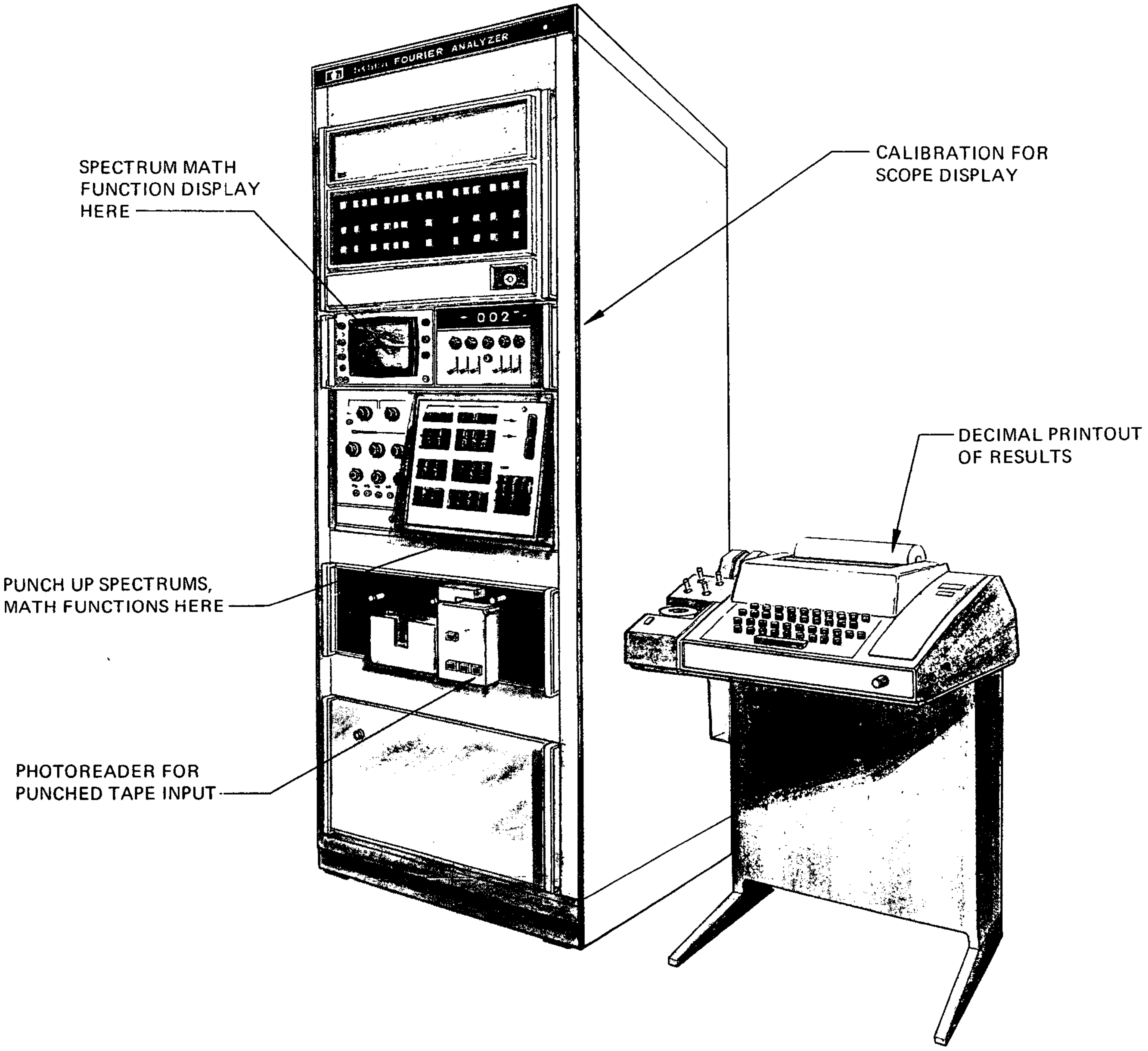
*HP's 8552/8553 Spectrum Analyzer – Frequency Components for Entire Scanwidth Are Displayed Simultaneously.*



Analog devices are slow at these low frequencies due to the physical characteristics of filters. When we get into math operations like averaging, correlation, convolution, power spectrum, etc., the analog instruments either can't do them, or if they can, they require cumbersome attachments, or a separate box for each function. On the Fourier Analyzer, the functions are available simply by pressing keys. Thus, the Fourier Analyzer is much more flexible than analog or special purpose systems. No special add-on's are needed for a wide variety of functions.

Finally, a selection of inputs and outputs are available which are not all possible with analog devices:

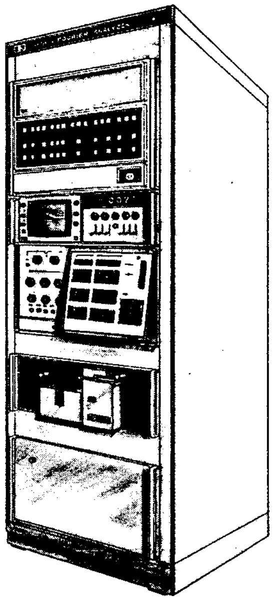
- Input — analog, paper tape, digital mag tape, manual entry of data via Keyboard, entry from remote computer
- Output — scope display, paper tape, digital mag tape, decimal number printout, external plotter, external scope, remote computer


FOURIER ANALYZER
002
SPECTRUM MATH FUNCTION DISPLAY HERE
CALIBRATION FOR SCOPE DISPLAY
DECIMAL PRINTOUT OF RESULTS
PUNCH UP SPECTRUMS, MATH FUNCTIONS HERE
PHOTOREADER FOR PUNCHED TAPE INPUT

# SECTION II

## How the Fourier Analyzer Works

## A Guided Tour of Some Basic Operations



## HOW FOURIER ANALYZER SYSTEM WORKS

As shown in Figure 2-2, data can be entered into the Fourier Analyzer:

- through the Analog-to-Digital Converter (ADC)
- as punched paper tape through the Photoreader
- from digital mag tape
- from digital mag disc
- manually by pressing keys on the Keyboard
- directly from a remote computer

Once the data is entered, it is stored in a data block in the Computer as shown in Figure 2-1. It is then ready for processing operations–Fourier transform, power spectrum, manipulations with data in other blocks, automatic routines, etc.–as called for from the Keyboard. The results of these operations are always displayed on the scope. In addition as shown in Figure 2-2 they can be:

- printed out in decimal numbers on the Teleprinter
- put on digital mag tape
- stored on mag disc
- sent back to a remote computer
- punched out on an optional paper tape punch
- plotted on an optional plotter
- displayed on an optional external scope

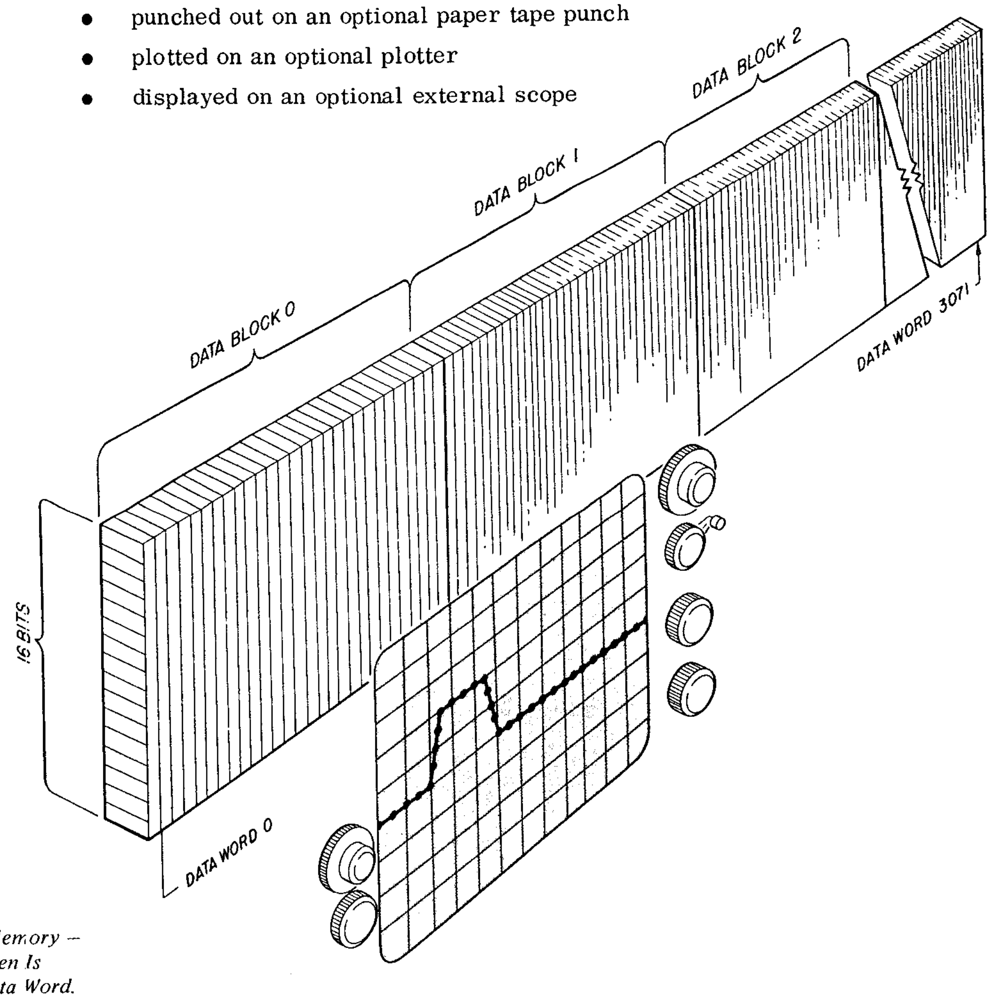


*Fig. 2-1 Concept of Analyzer Memory – Each Point on the Screen Is Determined by One Data Word. N Data Words = One Data Block. N Can Be 64, 128, 256 . . .*

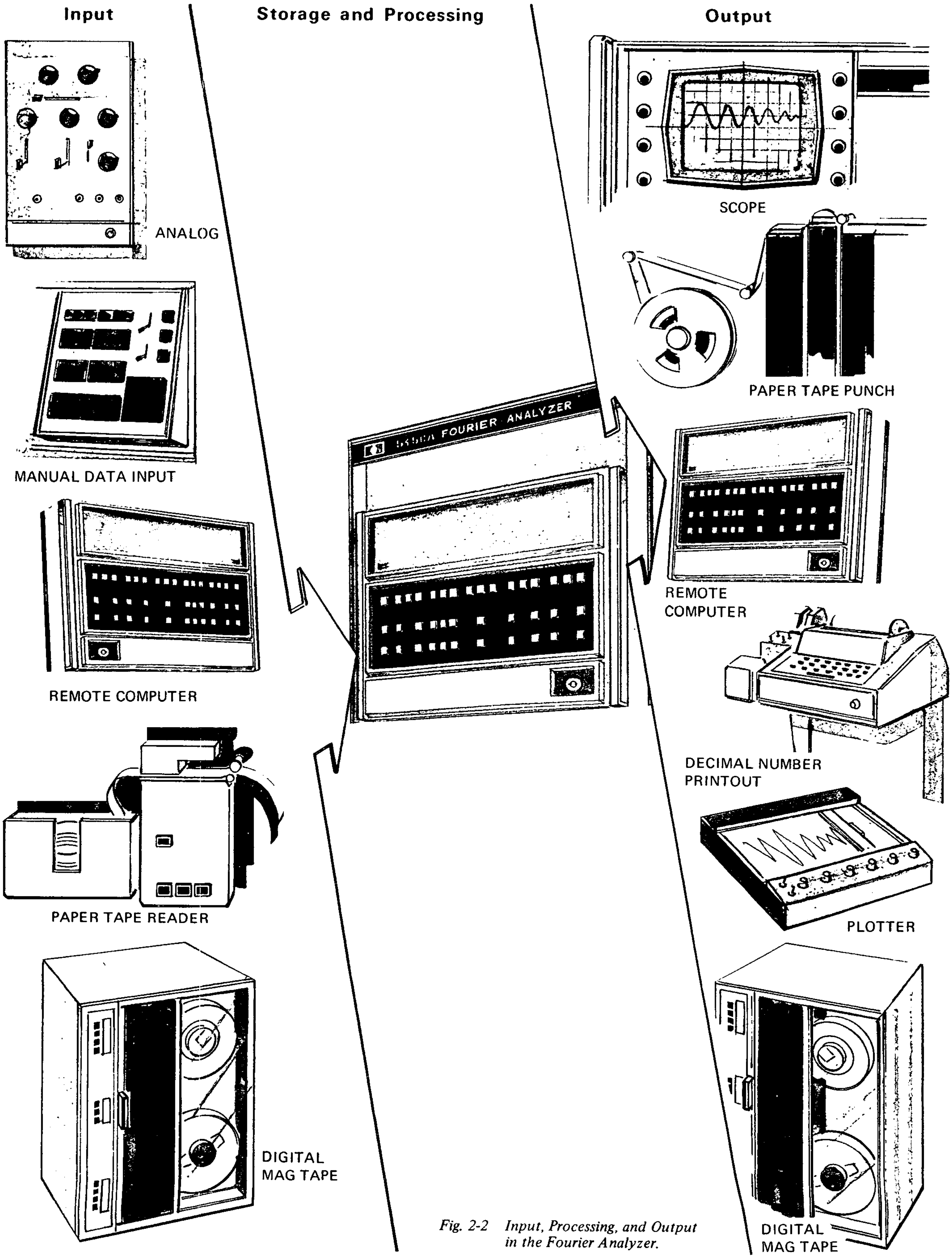


*Fig. 2-2 Input, Processing, and Output in the Fourier Analyzer.*

## HOW THE ANALOG-TO-DIGITAL CONVERTER WORKS

### Time Domain

As shown in Figure 2-3, the ADC samples the continuous analog input. Each sample becomes a digital word, stored in memory for later processing (e.g., conversion into the Fourier transform). The sampling parameters, or in other words, the time domain parameters are:

$\Delta t$ – the time between samples, called the "sample interval." ($\Delta$ TIME on the ADC panel).

N – the number of samples taken; this is the data block size (BLOCK SIZE on the Keyboard).

T – the total time of the sample record, also called "total record length". (TOTAL TIME on the ADC panel). From Figure 2-3 it can be seen that:

total record length = no. of samples x sample interval

$$T \quad = \quad N \quad \times \quad \Delta t$$

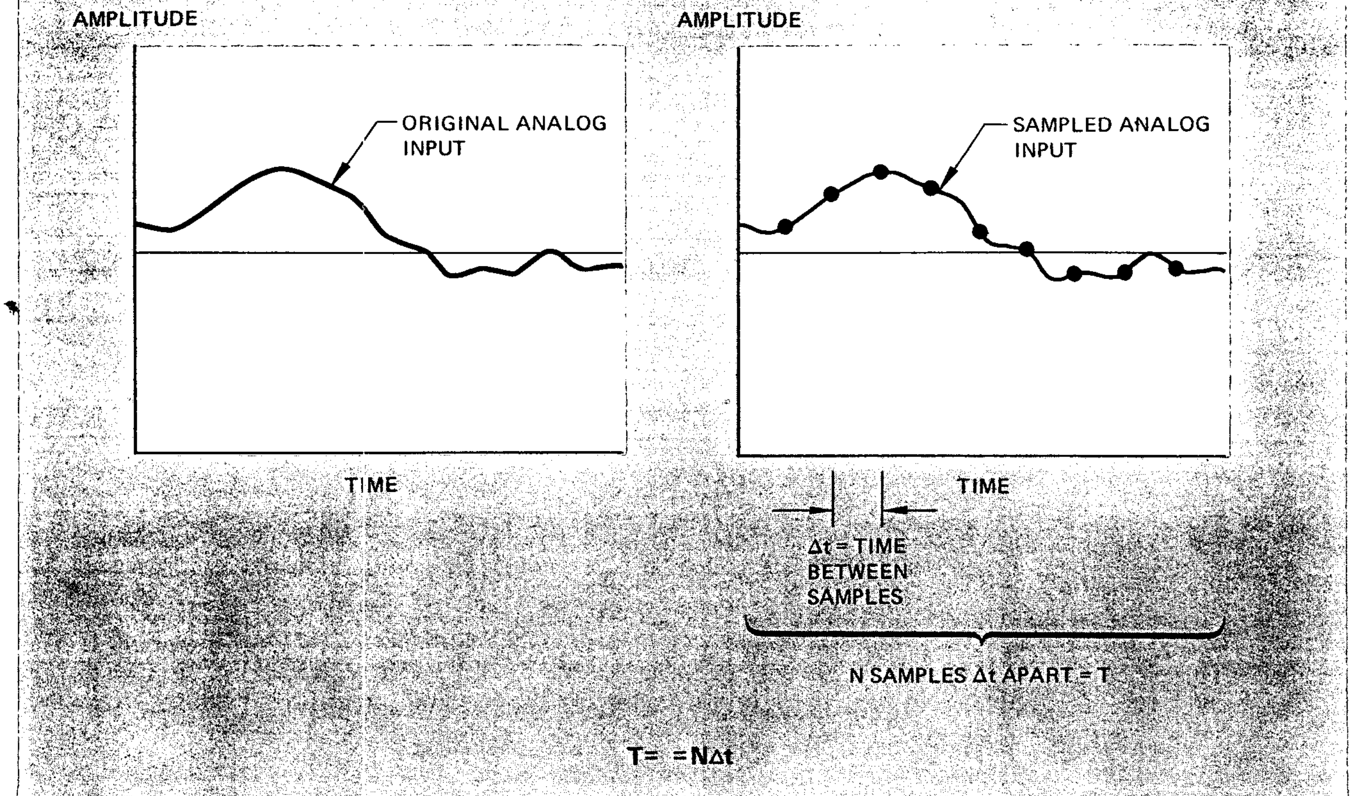
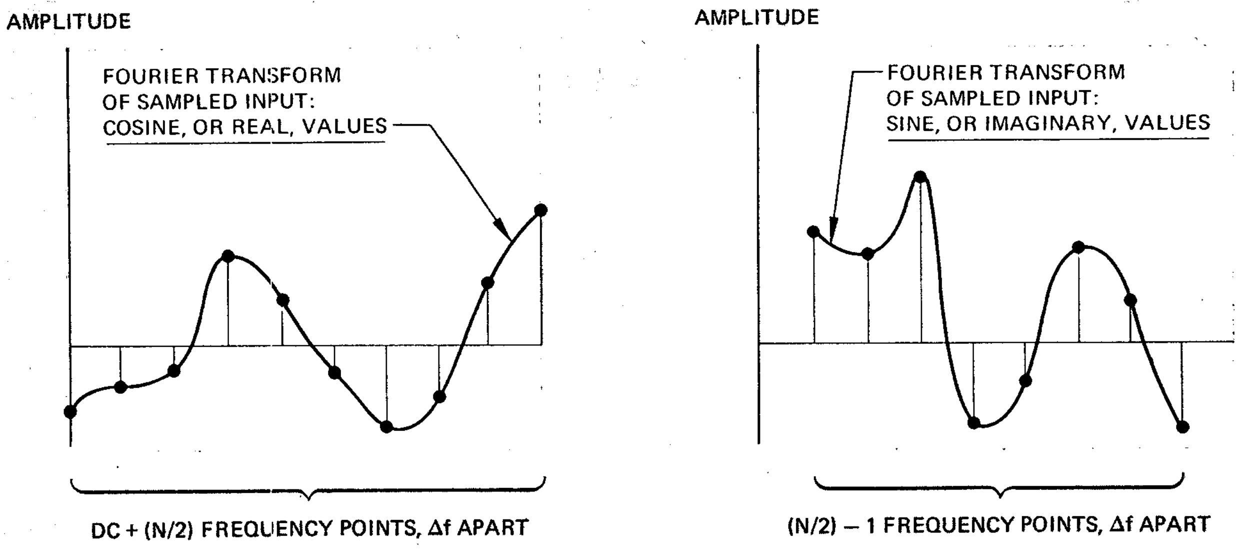


*Fig. 2-3 Time Domain.*



**Fmax = (N/2)(Δf)**

*Fig. 2-4 Frequency Domain.*

## Frequency Domain

Once we perform a Fourier transform, there is a similar set of parameters in the frequency domain, as shown in Figure 2-4.

Δf—the number of Hz between frequency points, or, as more familiarly known, the frequency resolution. Origin of display is 0Δf (DC component, on real (cosine) displays only); next point is 1Δf (fundamental frequency); next point is 2Δf (first harmonic); next 3Δf (second harmonic), etc. Frequencies between the harmonics will not show. To make them show, a smaller Δf must be used (but there are limits to this, as explained on page 2-6 ). Δf on the ADC panel is called ΔFREQ.

N/2–the number of frequency points: this is half the block size, or N/2, because the frequency information is broken down into two displays; real or magnitude (depending on MODE switch setting), and imaginary or phase (depending on MODE switch setting).

$F_{max}$—the maximum frequency of the display, or in other words, the bandwidth. (MAX FREQ on the ADC panel.) From Figure 2-4, it can be seen that:

$$\text{maximum frequency} = \text{no. of frequency points} \times \begin{matrix}\text{frequency}\\ \text{resolution}\end{matrix}$$

$$F_{max} \quad = \quad N/2 \quad \times \quad \Delta f$$

The time and frequency domains are related as shown in Figure 2-5.

sample interval = reciprocal of 2 times the maximum frequency

$$\Delta t = \frac{1}{2F_{max}}$$

frequency resolution = reciprocal of total record length

$$\Delta f = \frac{1}{T}$$

This means that changing one parameter will change the others.

Table 2-1 summarizes the equations above, and is also given in the Fourier Analyzer operating manual so that the user can obtain the best trade-off on the parameters he is interested in.

For example, suppose the user must have a 1 Hz frequency resolution ($\Delta f$) and at the same time wants a 5 kHz maximum frequency ($F_{max}$). He goes into Table 2-1 at line 3.

$$\Delta f = 1$$

In the last column, he sees that the equation relating frequency resolution ($\Delta f$) and maximum frequency ($F_{max}$) is:

$$F_{max} = (N/2)\ \Delta f$$

So:

$$5000 = (N/2) \times 1$$

$$N = 10,000$$

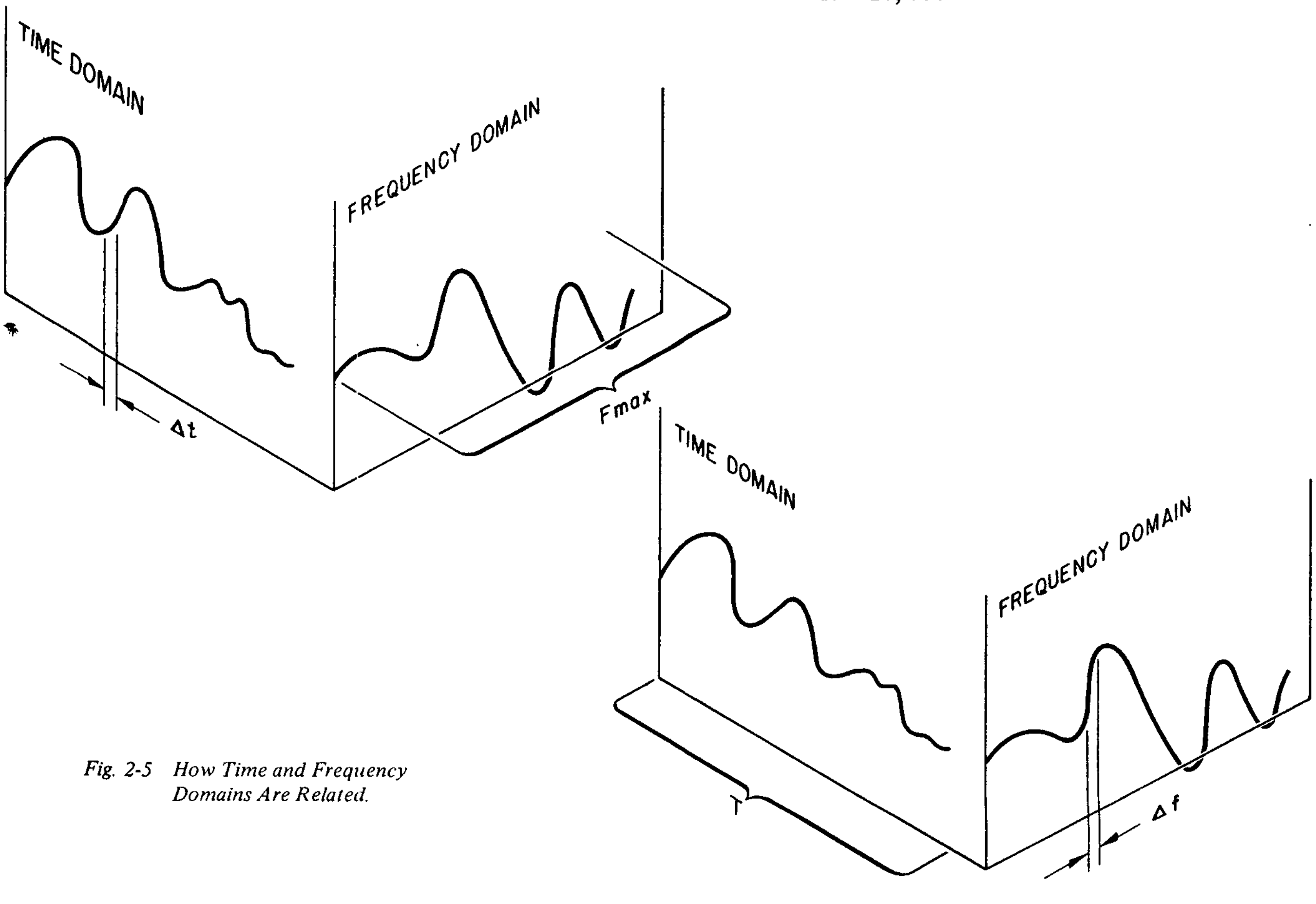


*Fig. 2-5 How Time and Frequency Domains Are Related.*

But the largest block size in the 8K Fourier Analyzer is 1,024, and in the 16K is 4,096. So a block size of 10,000 is impossible. Something has to give. Suppose the user is willing to settle for a lower maximum frequency. Assuming he has an 8K machine, he will enter the largest block size available, 1,024, in the $F_{max}$ equation, because this will give him the largest possible $F_{max}$:

$$F_{max} = 1,024/2 \times 1$$

$$F_{max} = 512 \text{ Hz}$$

If the user wants a 1 Hz resolution on an 8K machine, he must settle for a 512 Hz maximum frequency. Of course, if he needed the 5 kHz maximum frequency, he could have obtained it at the expense of some frequency resolution.

This is the kind of manipulation of ADC parameters which the user must be able to do. The parameters are set with the SAMPLE MODE and MULTIPLIER switches on the ADC, plus the BLOCK SIZE key on the Keyboard.

Table 2-1. Selecting Values for Data Sampling Parameters

| Choose convenient round number for parameter shown. | Chosen parameter automatically fixes the value of parameter below, because of relationship in parentheses. | Then make either of the remaining two parameters (can't be both) as close as possible to the desired value by choosing N* in the relationships shown. |
|---|---|---|
| 1. $\Delta t$ | $F_{max}\left(F_{max} = \frac{1}{2\Delta t}\right)$ | $T\ (T = N\Delta t)$<br>$\Delta f\left(\Delta f = \frac{1}{N\Delta t}\right)$ |
| 2. $F_{max}$ | $\Delta t\left(\Delta t = \frac{1}{2F_{max}}\right)$ | $T\ (T = N\Delta t)$<br>$\Delta f\left(\Delta f = \frac{1}{N\Delta t}\right)$ |
| 3. $\Delta f$ | $T\left(T = \frac{1}{\Delta f}\right)$ | $\Delta t\left(\Delta t = \frac{T}{N}\right)$<br>$F_{max}\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$ |
| 4. $T$ | $\Delta f\left(\Delta f = \frac{1}{T}\right)$ | $\Delta t\left(\Delta t = \frac{T}{N}\right)$<br>$F_{max}\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$ |

*N, the data block size, is always a power of 2.

## A GUIDED TOUR OF SOME KEYBOARD FUNCTIONS

The following guided tour explains some of the main concepts involved in using the Fourier Analyzer. It does not cover all keys and switch settings, since doing so at this point would obscure the presentation of essential ideas.

### Coordinates

Since every display on the Fourier Analyzer is in one set of coordinates or another, it is best that we understand these right at the outset.

In Figure 2-6 is a rectangular pulse and its Fourier transform, consisting of a real, an imaginary, and a complex display. Each of these displays is actually a different view of the same thing, as shown in Figures 2-6 and 2-7. The spiral actually curves around the frequency axis, as shown in the complex plot in Figure 2-7, which is an end view.

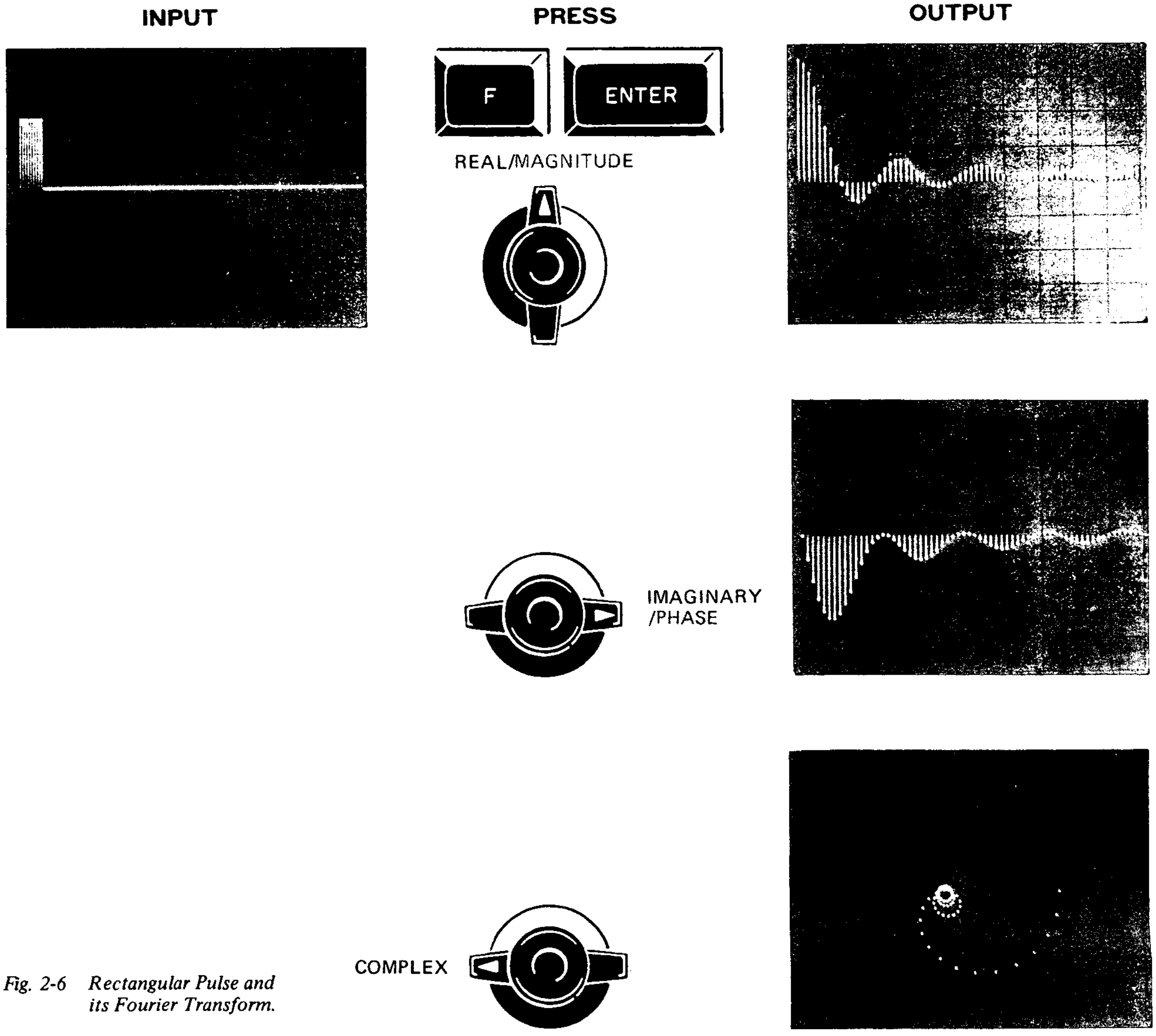


*Fig. 2-6 Rectangular Pulse and its Fourier Transform.*

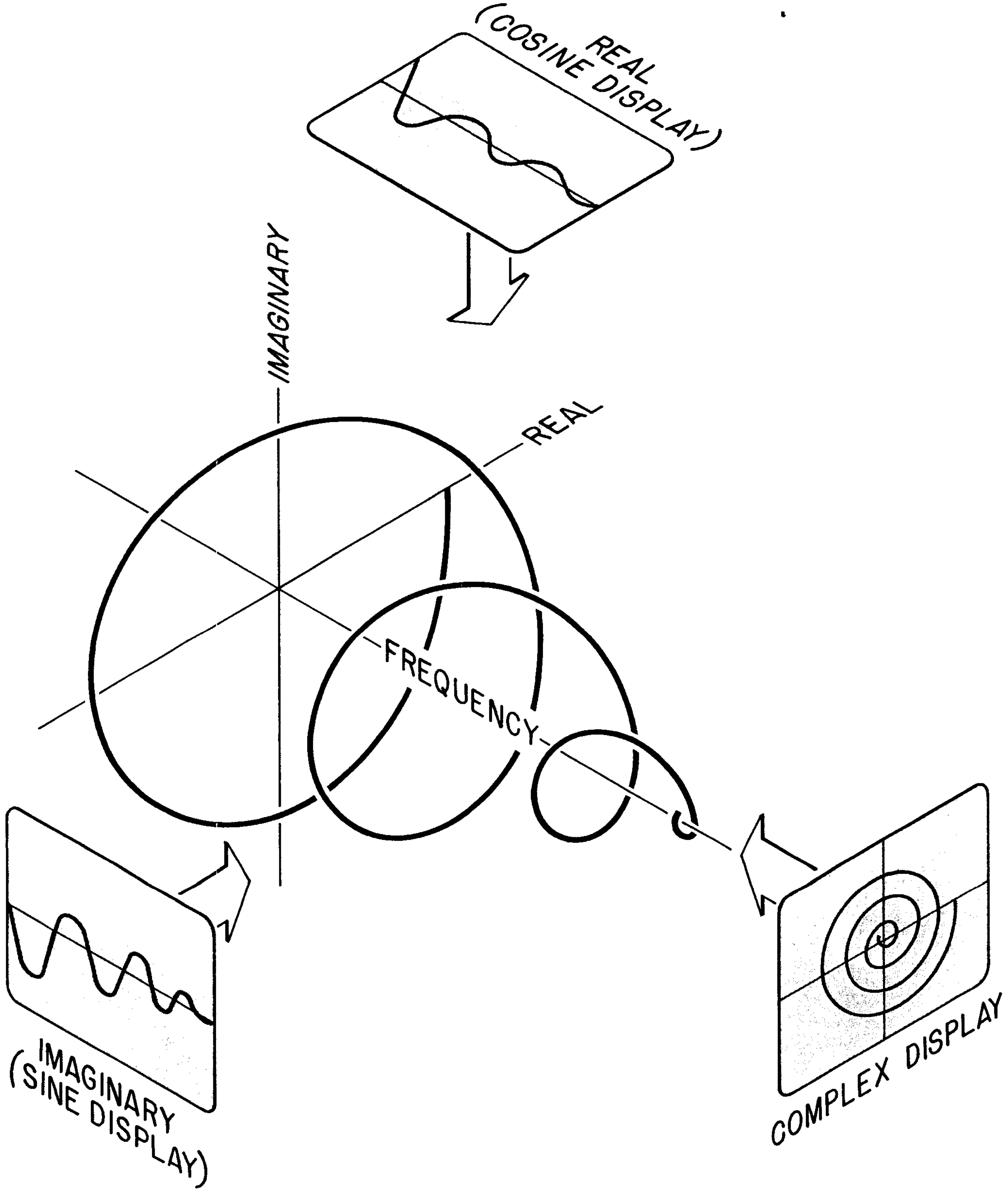


*Fig. 2-7 Fourier Transform of Rectangular Pulse.*

Rectangular coordinates, in which the machine normally operates, look at the real and imaginary planes cut by the spiral. Of course, other Fourier transforms are not spirals–only those for a rectangular pulse–but the arrangement of planes is always the same. Another view of the situation, taking one frequency point only, is given in Figure 2-8.

Polar coordinates are obtained by pressing POLAR then ENTER, and they are illustrated, for the same frequency point, in Figure 2-9. Here we are looking at the radii of the spiral (the lines). If we flip the MODE switch to phase, then their angle, or the phase of the frequency point, is shown.

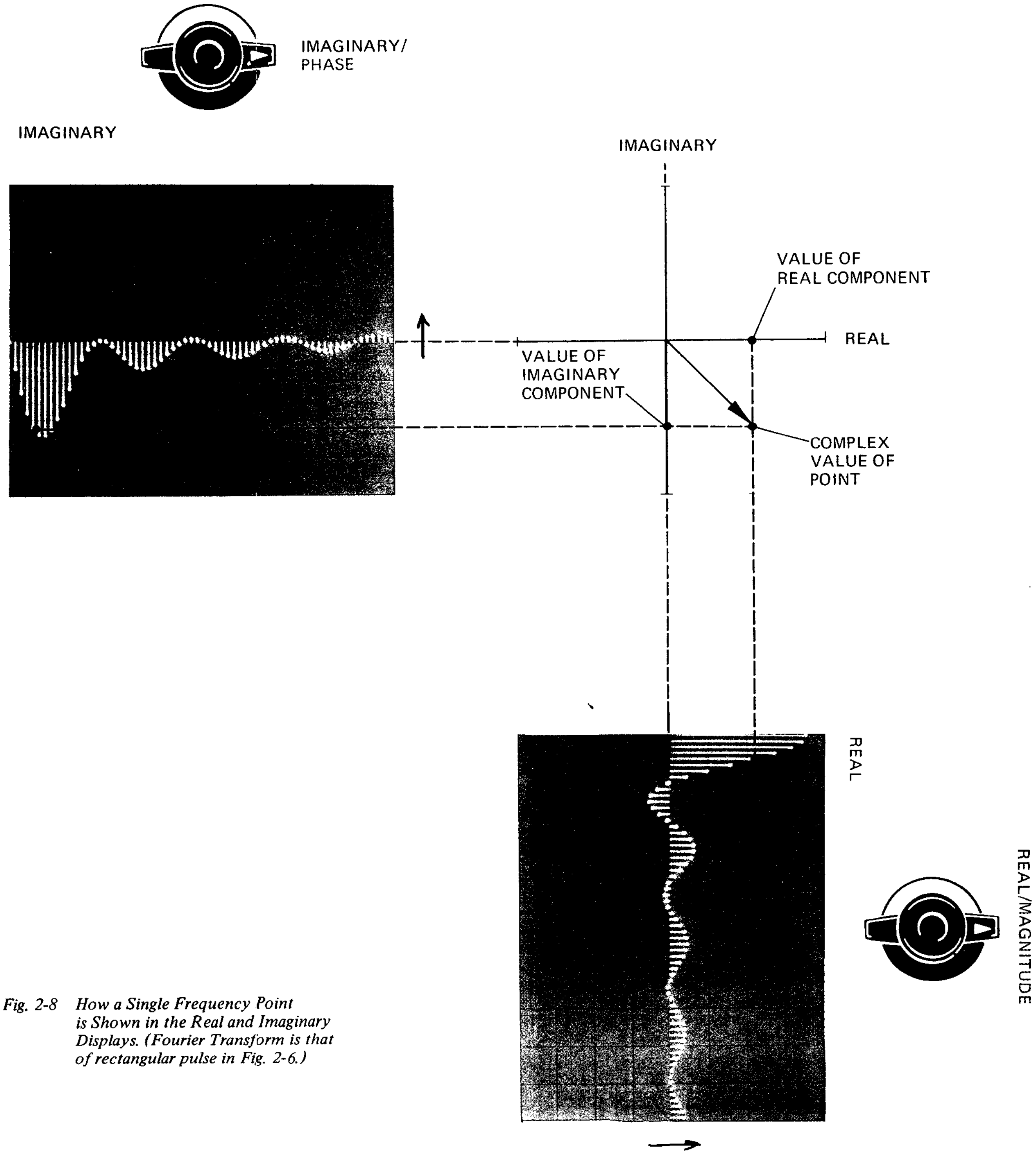


*Fig. 2-8 How a Single Frequency Point is Shown in the Real and Imaginary Displays. (Fourier Transform is that of rectangular pulse in Fig. 2-6.)*

The user chooses rectangular or polar coordinates depending on his requirements for the particular experiment he is running. Needless to say, both always show the same information.

To get back from polar to rectangular coordinates, the user presses RECT and ENTER.

With the coordinate sets behind us, we are now in a position to consider some of the basic keyboard functions.

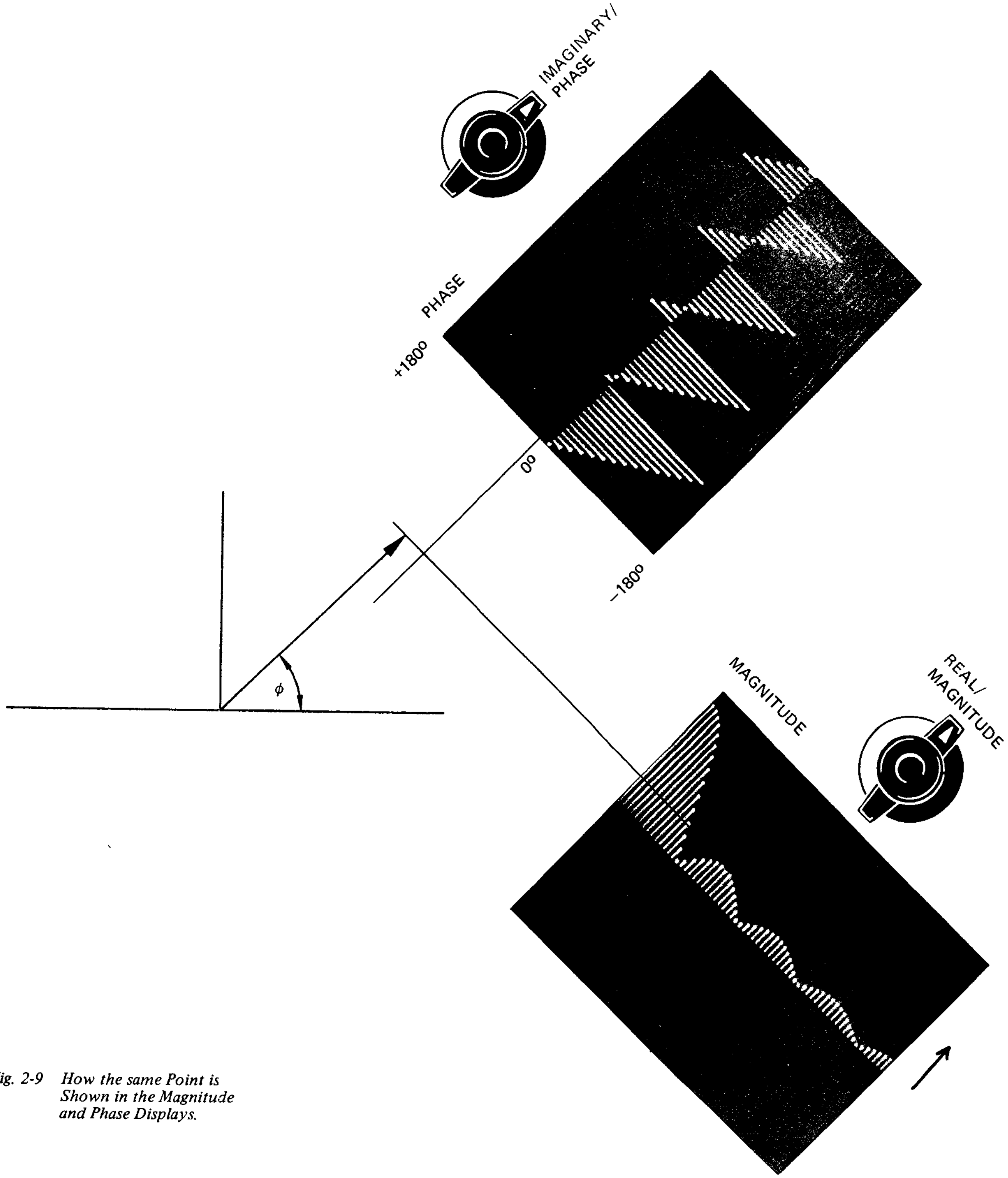


*Fig. 2-9 How the same Point is Shown in the Magnitude and Phase Displays.*

### DC Input

A DC input in the time domain appears as shown in Figure 2-10. If the user then presses F and ENTER, the Fourier transform of the DC input will appear. It is a single vertical line on the zero axis of the real (cosine) frequency display. The transform of the DC input is not shown on the imaginary display–that is, the zero axis of the imaginary display is always 0.

### Sine Wave Input

If the user has a pure sine wave input, as shown in Figure 2-10, and again goes for the Fourier transform by pressing F ENTER, he will get 0 for the real part display, and a single vertical line, either above or below the horizontal axis, for the imaginary part. (This assumes that the input sine wave is "periodic in the sampling window", a concept explained in Section 3.) If part of the sine wave is cut off, as viewed on the scope, then the user gets some smaller lines on the side of the main line in the imaginary display, and possibly some additional lines in the real display. Section 3 has the story on this.

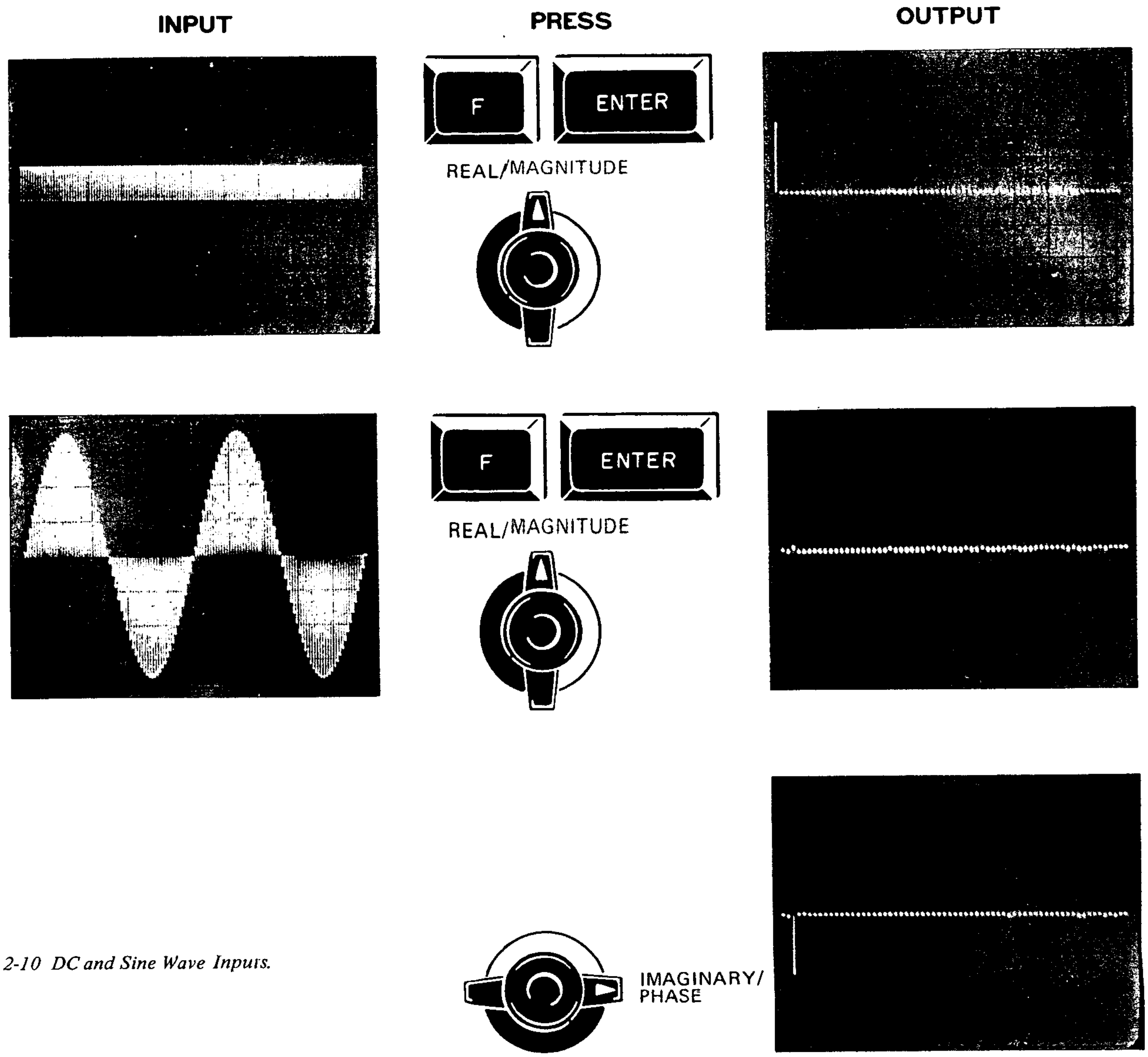


*Fig. 2-10 DC and Sine Wave Inputs.*

All sine (imaginary) lines are 180° out of phase with the original input, because of the nature of the Fourier transform. This is why the line points down, instead of up, in Figure 2-10.

### Cosine Wave Input

A pure cosine wave input, shown in Figure 2-11, gives a single vertical line for the real part display (in phase with the original input, hence pointing up in the Figure), and 0 for the imaginary part.

### Rectangular Pulse Input

Now let us take a look at the rectangular pulse input we used to explain the coordinate systems. Figure 2-11 shows the pulse. Press F and ENTER and we get the familiar attenuated sinusoidal curve (which is actually the curve: (sin x)/x, = sync x). Switch to imaginary, we see the sine plane version. Switch to complex, and we see the end view of the spiral, which we saw back in Figure 2-7.

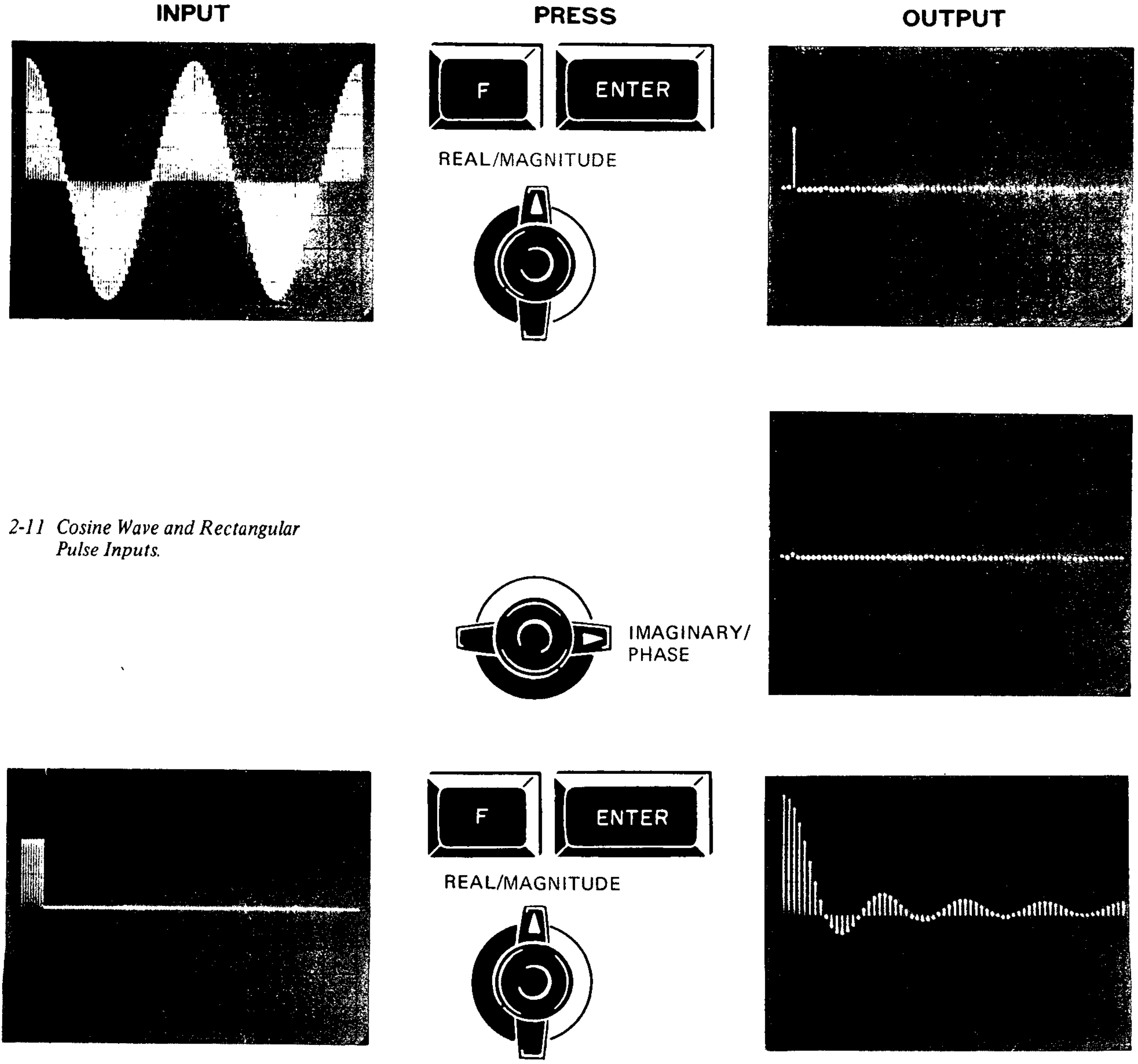


*Fig. 2-11 Cosine Wave and Rectangular Pulse Inputs.*

Press POLAR and ENTER and we see the magnitude display, and, switching to the phase display, the phase. (Understand that what this phase display shows is that the first point of the spiral is right on the cosine axis of Figure 2-7, the second point, (or line, or frequency) is a little below, the third a little more below, etc. around to -180°, whereupon the next loop of the spiral begins with the frequency line slightly above the cosine axis, then curves down and around as did the previous loop, etc.)

Suppose the user wants the vertical axis now to be logarithmic instead of linear. He presses LOG ENTER and there it is, as shown in Figure 2-13

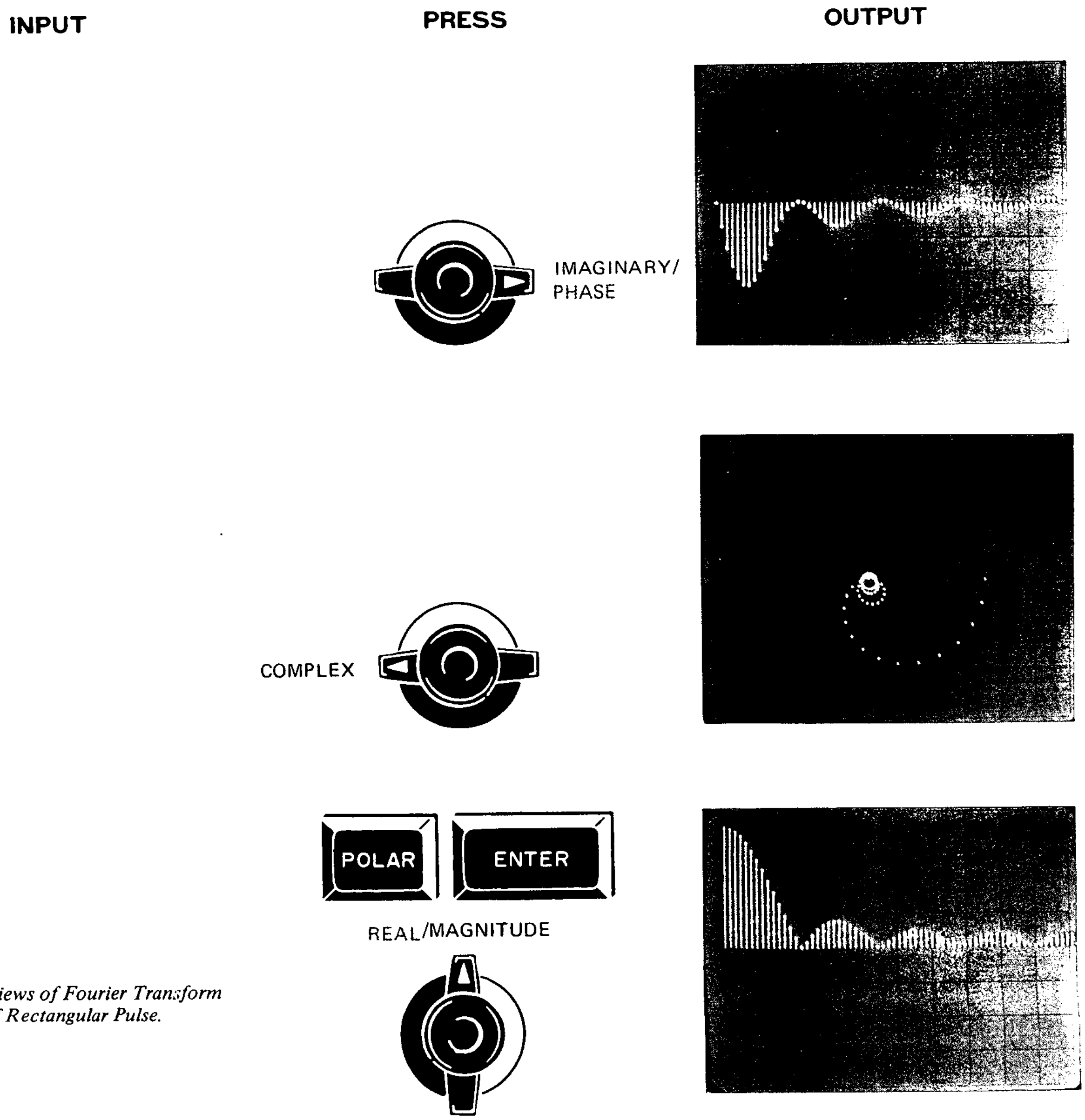


*Fig. 2-12 Views of Fourier Transform of Rectangular Pulse.*

(this converted the magnitude display to logarithmic in the vertical direction). If the user wants the horizontal axis to be logarithmic too, he sets a switch on the display panel and gets the display shown.

To get back out of the rectangular log, polar log, or polar linear display, he presses RECT then ENTER.

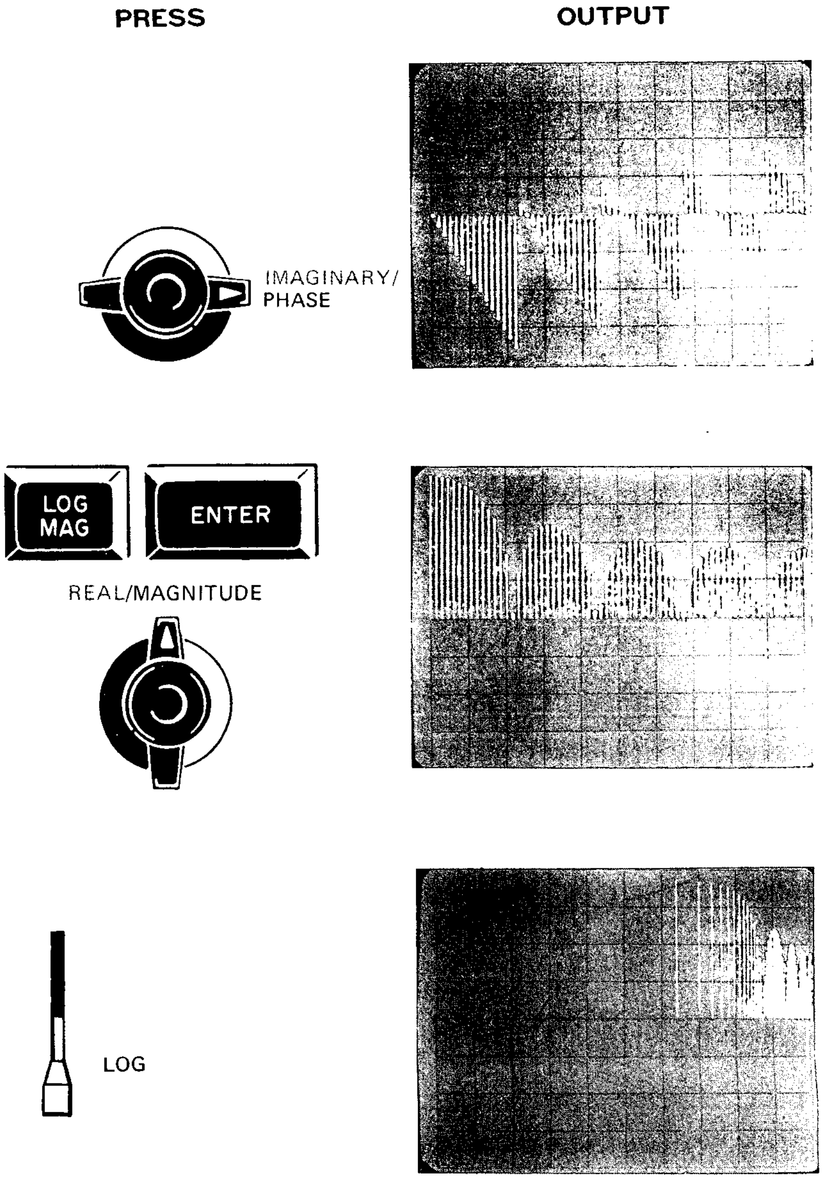


*Fig. 2-13 Further Views of Fourier Transform of Rectangular Pulse.*

## Power Spectrum

We come now to an important function for detecting signals in noise, and we will consider it at some length. Suppose a signal generator and a noise generator are hooked up as shown in Figure 2-14. The user obtains the Fourier transform of the combined signal and noise input by pressing F and ENTER, as shown in Figure 2-15. Then he presses MULT* and ENTER. (MULT* is the conjugate multiplication key, which is explained in Section 3.) The result is called the power spectrum, and, as can be seen, it may have a peak at the frequency of the signal, if the amplitude of the signal is large compared to the amplitude of the noise.

In real life, of course, the signal may be very small, and would not appear above the noise, This is where the programming abilities of the Keyboard come into prominence. Repeated samples of the input can be taken and the power spectrums can be added together over and over again so that eventually the signal, which is always at the one frequency, rises above the noise, which is random for all frequencies.

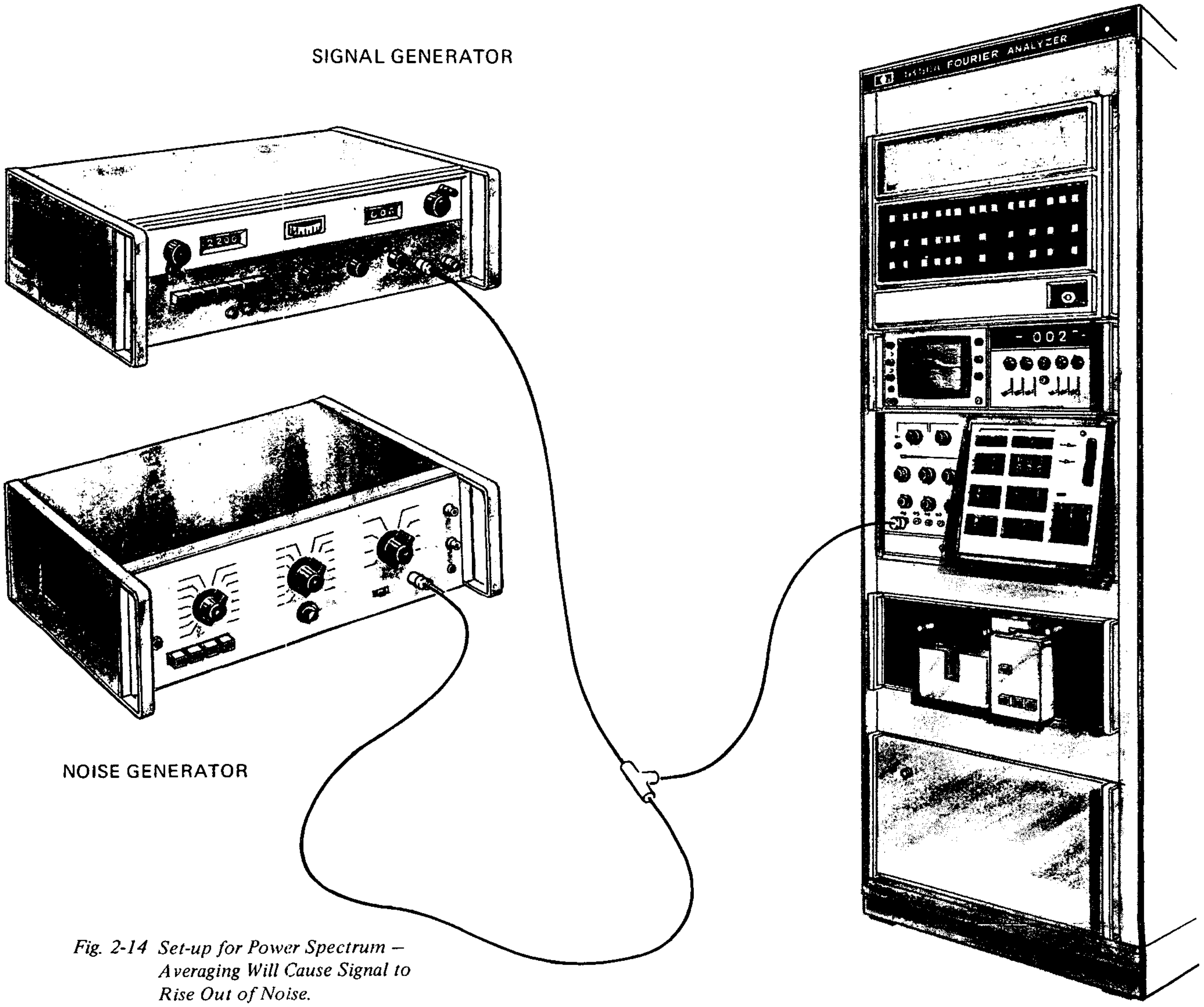


*Fig. 2-14 Set-up for Power Spectrum – Averaging Will Cause Signal to Rise Out of Noise.*

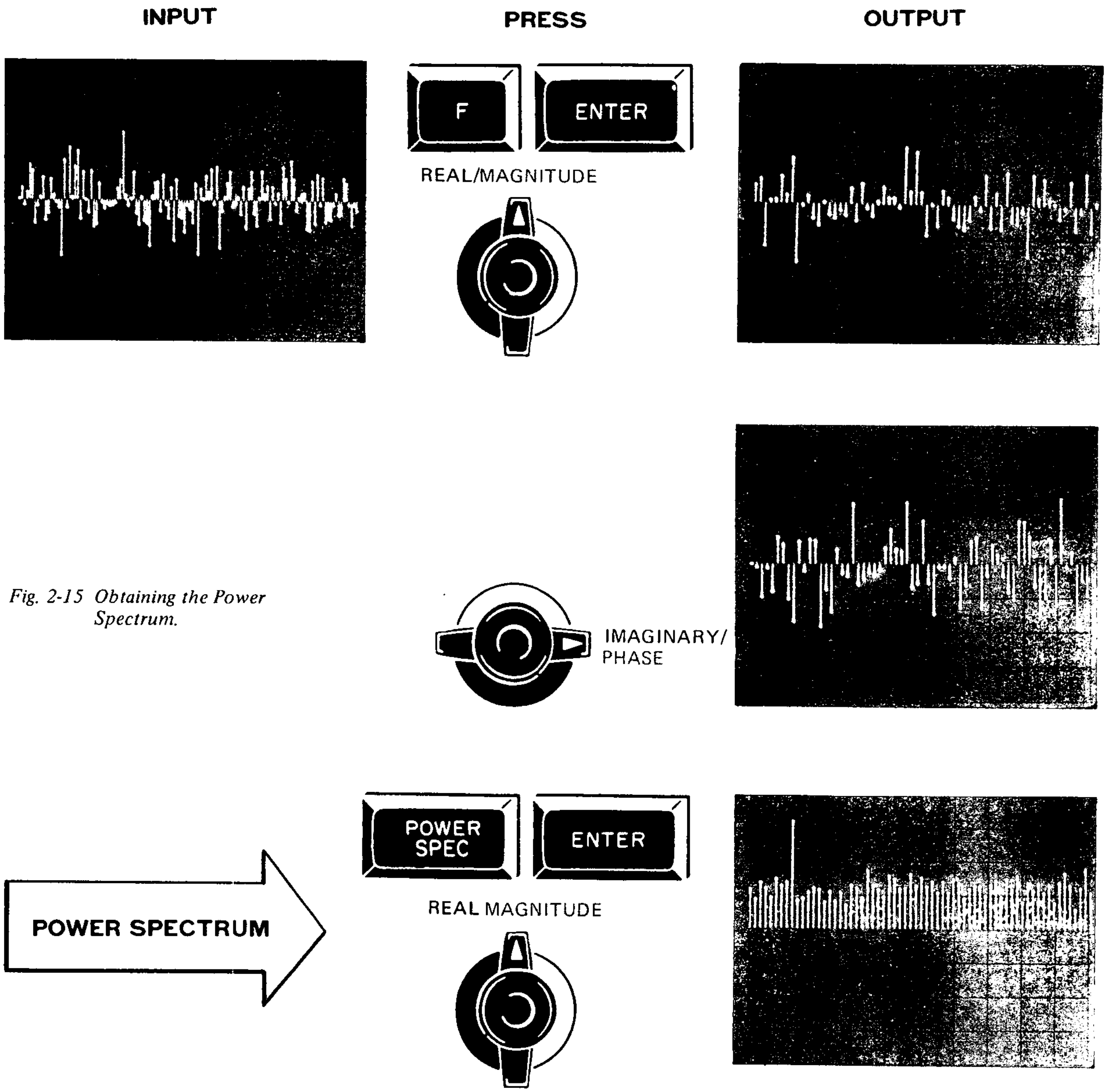


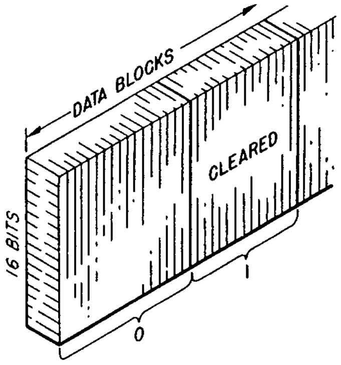
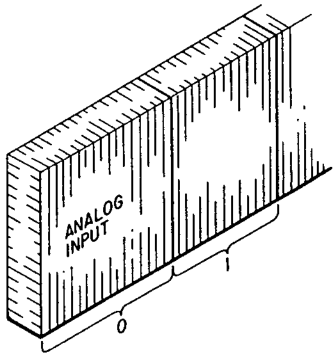
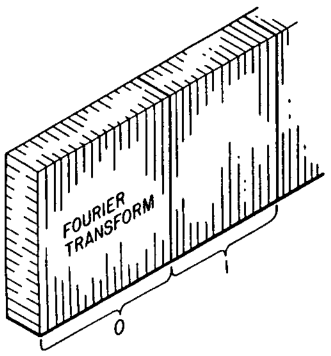
*Fig. 2-15 Obtaining the Power Spectrum.*

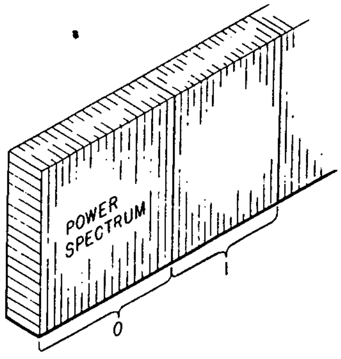
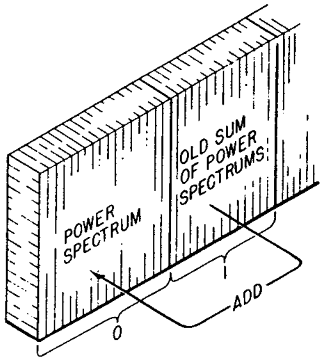
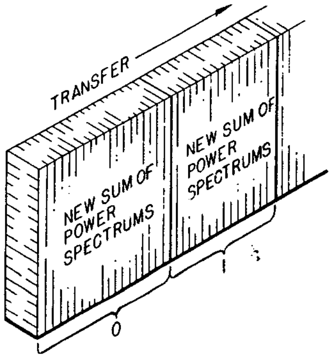
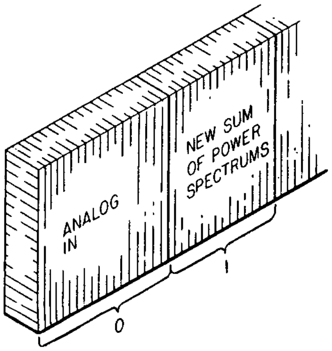
### The Power Spectrum Program

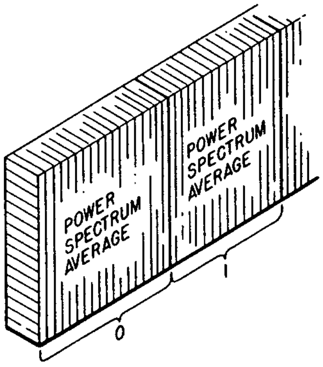
The next three pages show the keys the user would press, what happens, and how this appears in the data memory, in developing a programmed, averaged power spectrum. In brief, what the program is doing, is this:

1. taking N samples of the data (i. e., one record) and storing it in a data block.
2. doing a Fourier transform on this data.
3. conjugate multiplying the transform of this data.
4. adding the resulting power spectrum to the previous sums of power spectrums in another data block.
5. repeating this 32 times, then dividing the final power spectrum by 32, to get the average.
6. displaying the result.

After such a process, even a weak signal will rise above the surrounding noise, as shown in Figure 2-17, and be detectable.
Before running the program, refer to instructions on page 2-20.

<table>
<tr><th>COMMAND</th><th>MEANING</th><th>ACTION IN DATA MEMORY</th></tr>
<tr><td>RPLAC 0 ENTER<br>CLEAR 1 ENTER<br>LABEL 1 ENTER</td><td>Replace any existing commands in the program memory with the following program.<br><br>Clear block 1 of any residual data.<br><br>This allows the starting point of the program to be identified.</td><td>
</td></tr>
<tr><td>ANALG IN 0 SPACE<br>1 ENTER</td><td>Analog input (noise plus possible signal) to data block 0, display data block 1. Samples of the input will then be taken at intervals set by controls on the Analog-to-Digital Converter (ADC), converted to digital words, and sent to the data memory (block 0) for processing per succeeding commands. Block 1 will be displayed because it is the block where the cumulative sum of previous power spectrums will be stored. User can thus observe sum building up.</td><td>
</td></tr>
<tr><td>F 0 ENTER</td><td>Take the Fourier transform of block 0. The result is a set of real (cosine) and a set of imaginary (sine) values in the frequency domain.</td><td>
</td></tr>
</table>

| COMMAND | MEANING | ACTION IN DATA MEMORY |
|---|---|---|
| POWER SPEC ENTER | Pushing power spectrum * performs the following:<br>Conjugate multiply block 0 with itself. Each spectral line is thus multiplied by its own complex conjugate, making each spectral line's value equal to the power at that frequency.<br>Power spectrum = $(a_n + ib_n)(a_n - ib_n)$ |  |
| | Add block 0 to block 1. This adds the data in the current power spectrum, stored in block 0, to the sum of all past power spectrums, stored in block 1. The new sum is stored in block 0. |  POWER SPECTRUM<br>OLD SUM OF POWER SPECTRUMS<br>ADD<br>0<br>1 |
| | Transfer the result of the addition to block 1. On the next pass, this sum will be added to the new spectrum in block 0. |  TRANSFER<br>NEW SUM OF POWER SPECTRUMS<br>NEW SUM OF POWER SPECTRUMS<br>0<br>1 |
| COUNT 1 SPACE<br>3 2 ENTER | This step directs the program back to label 1 so that another record can be taken from the ADC, the power spectrum computed and added to block 1. This process is repeated 32 times. |  |

| COMMAND | MEANING | ACTION IN DATA MEMORY |
|---|---|---|
| ÷ 1 SPACE<br>3 2 ENTER | After the STORE command, two steps previous, the data in block 0 and block 1 are the same (a characteristic of how the data is handled).<br>The ÷ command divides the data in block 0 by 32 to yield the average value. (Data in block 0 and block 1 are of course the final cumulative sum of the power spectrums. The result of the division, i.e., the average power spectrum, is displayed on the scope.) |  |
| END ENTER | This step identifies the end of the program to the Analyzer. | |
| TERM ENTER | Identifies the end of the programming mode of operation to the Analyzer. | |

Before the Power Spectrum program above is operated, the ADC controls should be set as required to handle the input signal. The SAMPLE CONTROL settings are determined by the signal's frequency content, and the OVERLOAD VOLTAGE is set so the OVERLOAD VOLTAGE indicator doesn't light when the Analog Input operation is performed.

To make a quick check of ADC switch settings, connect the signal and noise to the ADC input, set REPEAT/SINGLE to REPEAT, and press

ANALG IN ENTER

to begin continuous Analog Input operation. Change the ADC control settings, if required.

Set REPEAT/SINGLE to SINGLE. Press

JUMP 1 ENTER

to begin operation of the Power Spectrum program above. The program will run to completion.

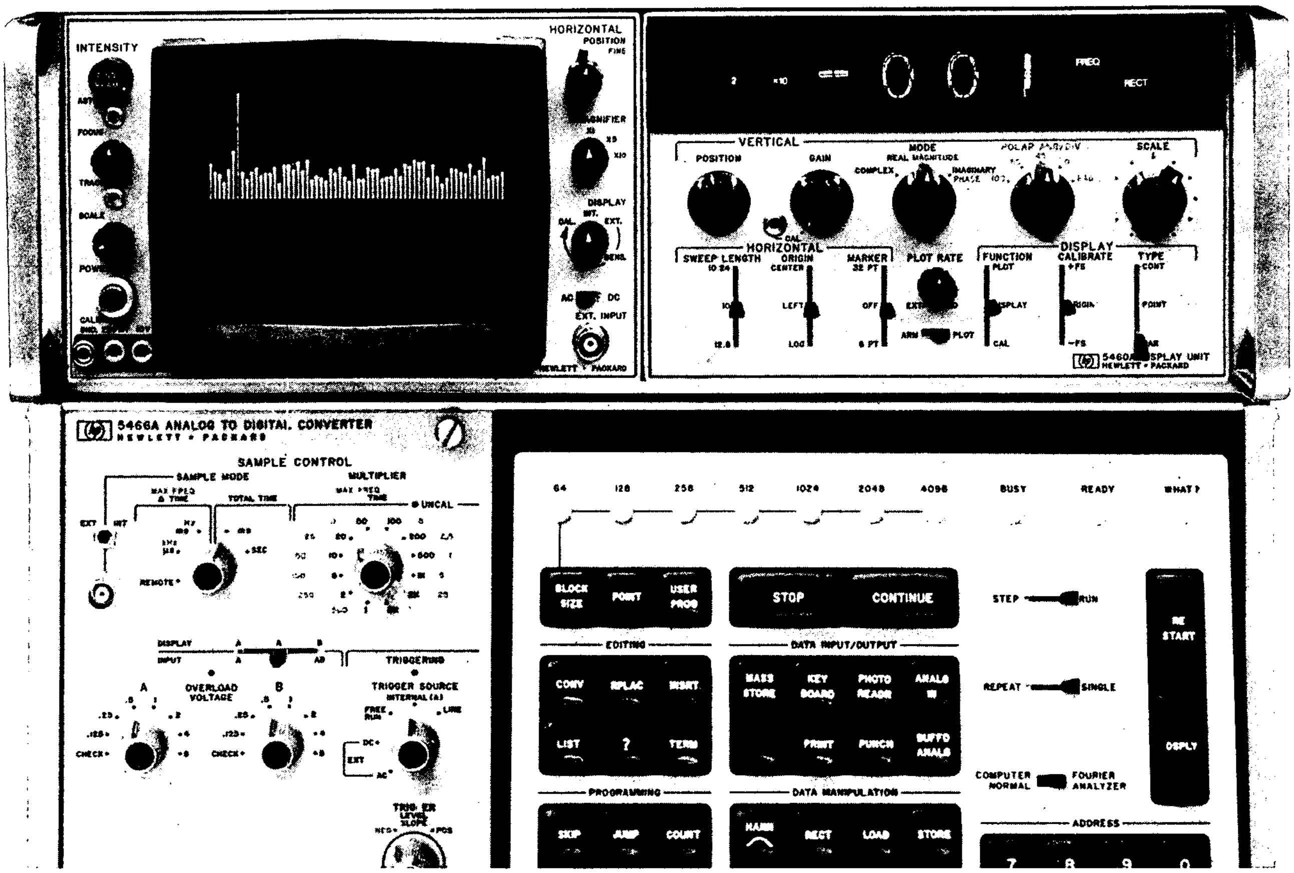


*Fig. 2-17 Power spectrum display.*

The program described on the previous pages produced the above display. The signal can clearly be seen above the noise.

The amplitude of the signal is read in accordance with the digital scale to the right of the scope. In this case, each vertical division on the scope is:

$$2 \times 10^{-001} = 2 \times 10^{-1} = 0.2 \text{ volts } \underline{\text{squared}}$$

(because this is a power spectrum)

Thus, we can see that the amplitude of the signal is about 0.7 volts squared. The SCALE switch on the upper right permits expanding or contracting of the display in the vertical direction: the digital scale changes with it so that the scope can always be read accurately.

Note that the digital display also shows we are looking at the FREQuency domain, in RECTangular coordinates.

The frequency is determined from the SAMPLE CONTROL switch settings on the ADC. SAMPLE MODE switch gives units; MULTIPLIER gives values. Units and values are tied together by color. Thus here, black "μs" and black "100" go together: this says that the input was sampled at 100 μs intervals (Δt = 100 μsec). Likewise, blue "kHz" and blue "5" on these same switch positions go together, meaning that the maximum frequency ($F_{max}$) is 5 kHz, i. e., the horizontal scale on the scope is 0 to 5 kHz.

Rough frequency approximation -- scope is 10 divisions across, signal is almost on first division means signal has approximate frequency of 5 kHz/10 = 500 Hz.

Precise frequency -- use equation from table on page 2-7 (line 3):

$$F_{max} = (N/2) \cdot \Delta f$$

$$\Delta f = \frac{F_{max}}{(N/2)}$$

Here we can see from the Keyboard that the N (BLOCK SIZE) chosen was 128, so

$$\Delta f = \frac{5000}{(128/2)} = \frac{5000}{64} = 78.0 \text{ Hz}$$

Now the first spike on the scope display is DC; counting beginning with the second, we see that the signal is the sixth spike. Six times 78.0 Hz is 468 Hz. Frequency of the signal is 468 Hz.

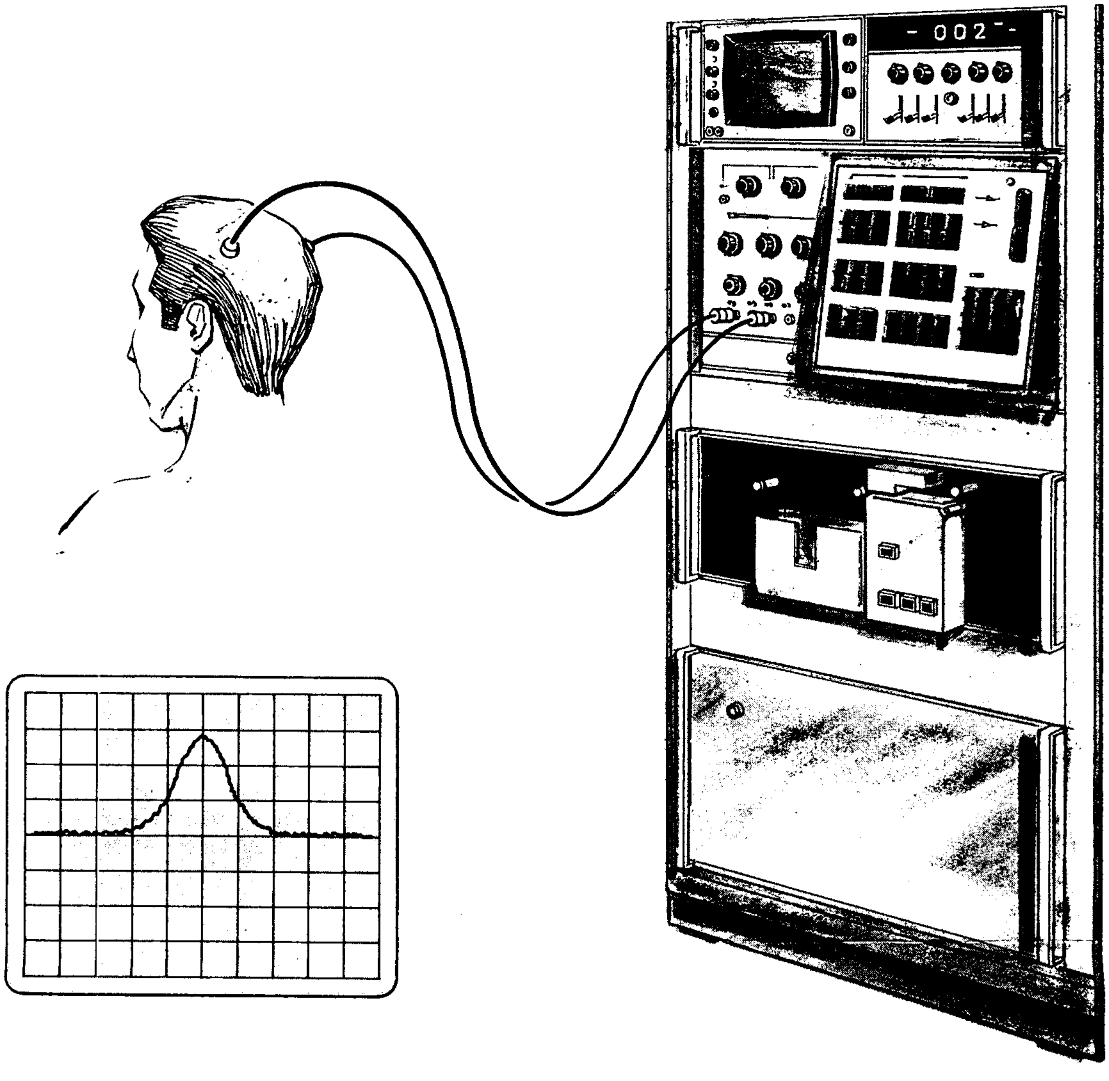
### How the Correlation Function Works

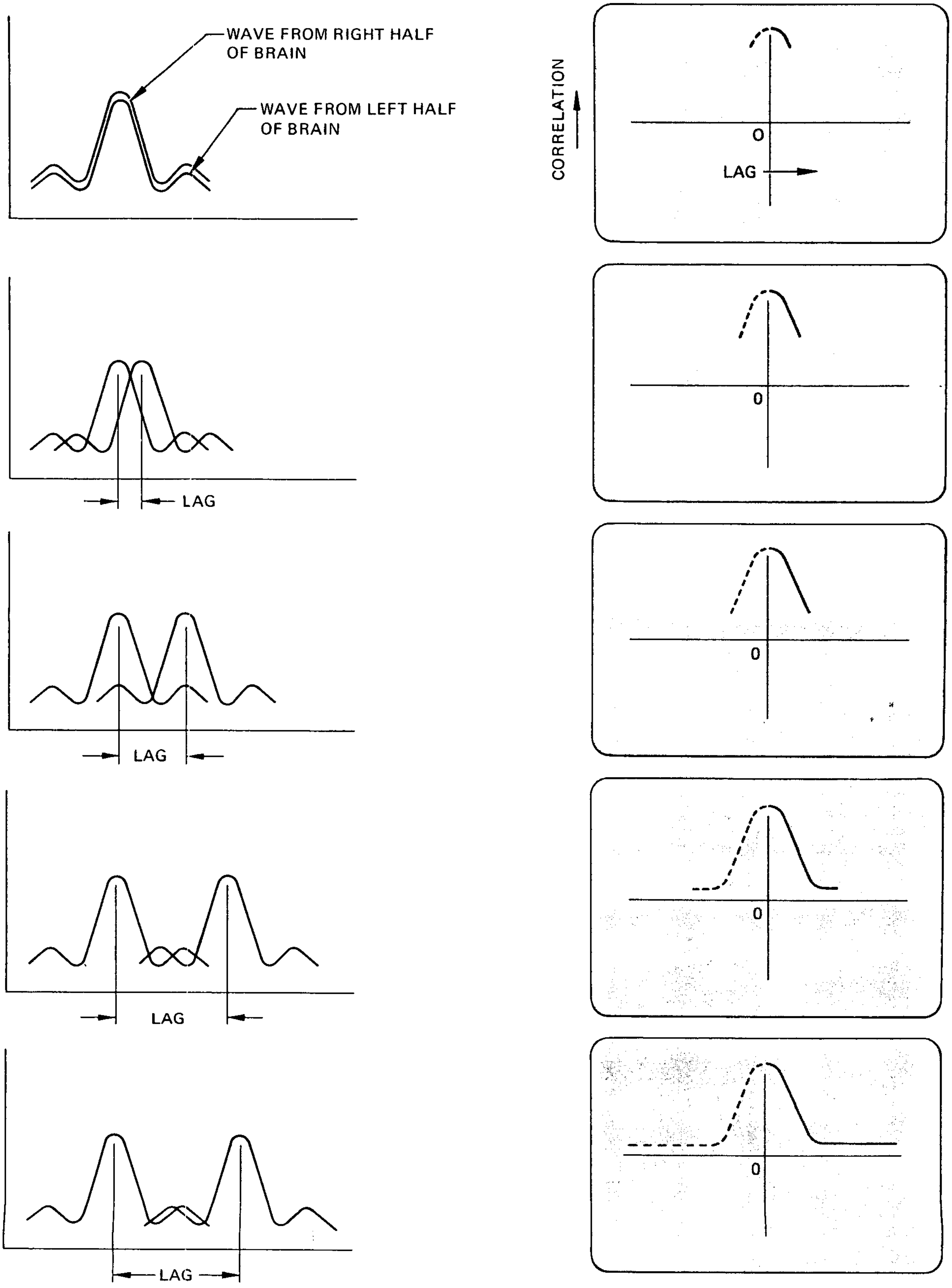
The Correlation Function presents the same information as the Power Spectrum; the difference is that Correlation is used in the time domain, while Power Spectrum is used in the frequency domain.

In the Power Spectrum example above, for instance, the final display (Figure 2-17). shows the frequency domain presentation of the signal-plus-noise input, and allows easy calculation of the single frequency of the signal. Correlation of this same input signal in the time domain would result in a display that would show a periodicity equal to the period of the signal (note that the correlation waveform would not necessarily be the same as the signal's waveform).

The two types of correlation are defined by the number of inputs being correlated. In auto-correlation, a single signal is applied to the correlator input(s) and with itself; any periodicity in the input would be indicated by a repetitive pattern, having the same period, in the display (see note in previous paragraph). In cross-correlation (as in cross power spectra), two separate signals are compared; the correlation function will indicate points of similarity.

As mentioned above, auto-correlation of a single input can be used to determine whether it contains any periodic components, and what the period of the signal is; for a single-frequency signal, the frequency can be calculated as 1/T, where T is the period. Cross correlation can be used to determine the time delay between two signals (for example, stimulus and response) in a noisy environment; this technique is being used extensively to gain insight into the functioning of the brain.





(CORRELATION FUNCTION IS SYMMETRICAL; MATHEMATICALLY, ONE SIDE IS COMPUTED FIRST, THEN SECOND) .

Power spectrum and correlation functions present the same information in two different ways:

| time domain | frequency domain |
|---|---|
| auto correlation (one input) | power spectrum |
| cross correlation (two inputs) | cross power spectrum |

There is a correlation key (CORR) that gives the auto or cross correlation function in one step, rather than going through the power spectrum. We have gone through the power spectrum by way of helping the understanding.

## Transfer Function

The transfer function is the mathematical description of a system, be that system a filter, an automobile engine, a vibrating airplane wing, an organ in the human body, or whatever. The transfer function can also be used to measure the relationship between any two signals. It can be defined as:

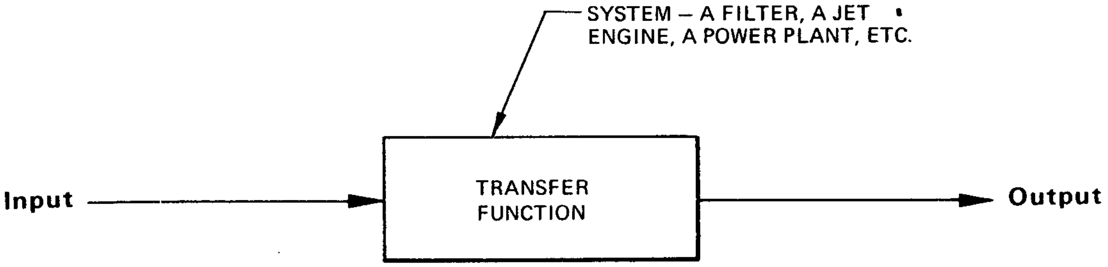
$$\text{Transfer function} = \frac{\text{Fourier transform of output}}{\text{Fourier transform of input}}$$

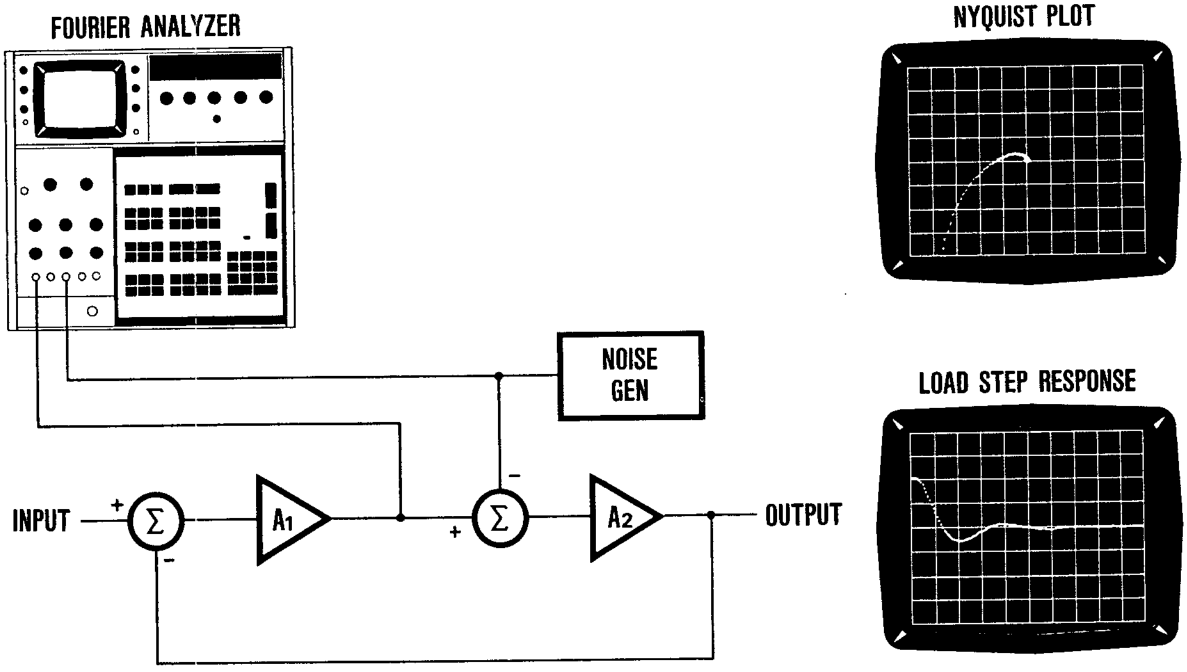
or equivalently:

$$\text{Transfer function} = \frac{\text{average cross power spectrum of input and output}}{\text{average power spectrum of input}}$$

Since phase information is important, the first method requires a time synchronization. For this reason, and because averaging gives a more reliable transfer function, the second method of calibrating the transfunction is more commonly used. A program to calculate the transfer function automatically, using the averaging method, can be entered via the keyboard's POWER SPECtrum and TRANSfer FunCtioN keys. Section 4 has more details on the mathematics of this function.

In real life, two techniques are used for obtaining the transfer function. In the first, the input is swept with a wide bandwidth of frequencies (input and output being plugged into the ADC as shown on page 2-25). This has the disadvantage of requiring that the system be shut down from its normal work for the purposes of the test–a distinct disadvantage if it is an aircraft or missile servo system being checked in flight. The second technique avoids this: Gaussian white noise of low amplitude is fed into the system while it is operating normally. (Again, system input and output are connected to the ADC as before.) Now the transfer function can be calculated or checked without interfering with the operation of the system.

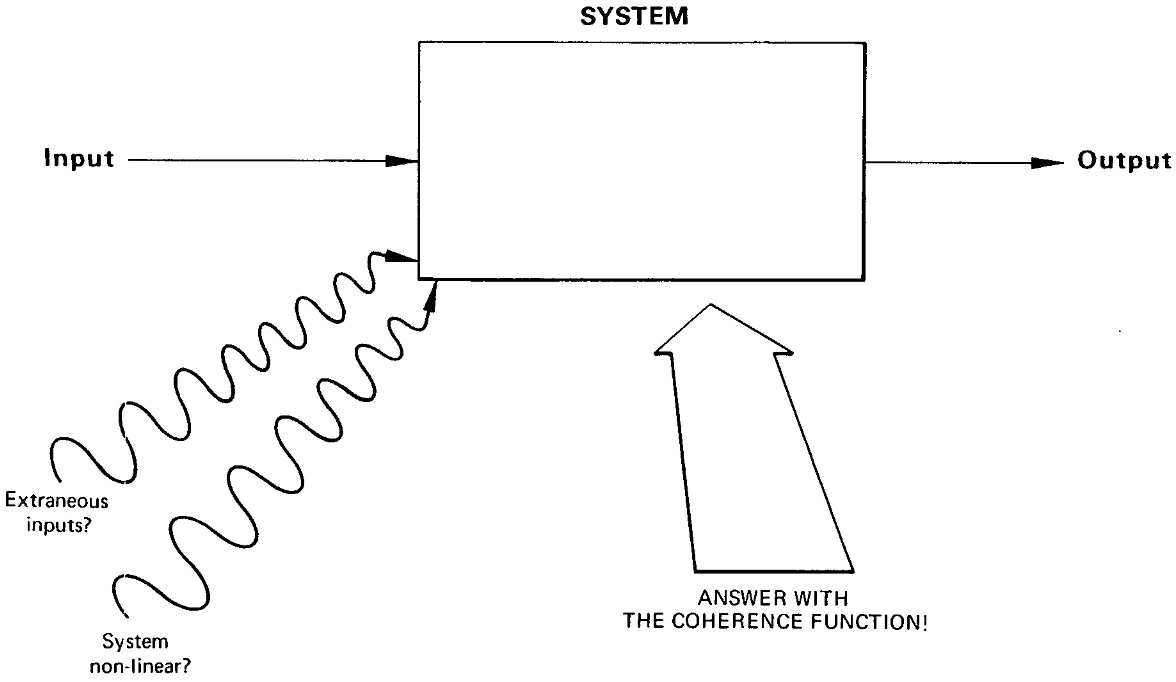

SYSTEM – A FILTER, A JET ENGINE, A POWER PLANT, ETC.
Input
TRANSFER FUNCTION
Output


FOURIER ANALYZER
NYQUIST PLOT
NOISE GEN
LOAD STEP RESPONSE
INPUT
+
Σ
−
A1
+
Σ
−
A2
OUTPUT

### Coherence Function

This can be used to check the validity of the transfer function. To put it another way: it can be used to measure the degree of causality between any two signals. The problem is this: when we compute a transfer function, we don't know if a) there were any extraneous inputs in the system, and b) if the system is linear or not. Both of these would place in error the transfer function computed.

Coherence function values range between "0" and "1". A "0" value means no coherence between input and output of the system (indicating that the system may have extraneous inputs or may not be linear). A coherence value of "1" means there is complete coherence between input and output (indicating that there is only one input and the system is linear).



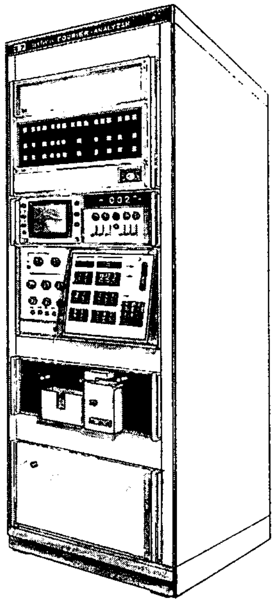
# SECTION III

## Sampling Window Error

## Aliasing

## Conjugate Multiplication

## Further Details on the N/2 Frequency Domain Points

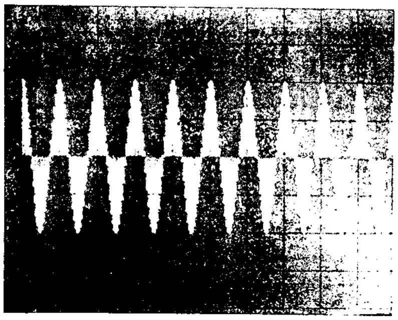


## SAMPLING WINDOW ERROR

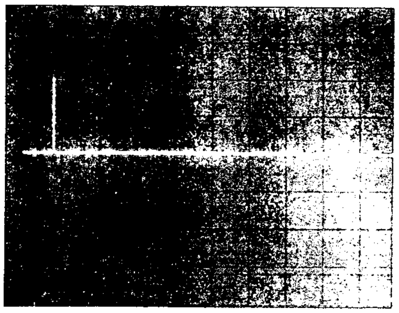
This results from an instrument, such as the Fourier Analyzer, assuming that whatever sample it takes is in fact the periodic function being studied. However, the transform will have erroneous amplitudes, plus side lobes which can conceal low amplitude signals if the sample "window" was not situated over the actual beginning and end of the periodic function.

FUNCTION PERIODIC
IN SAMPLING WINDOW

INPUT

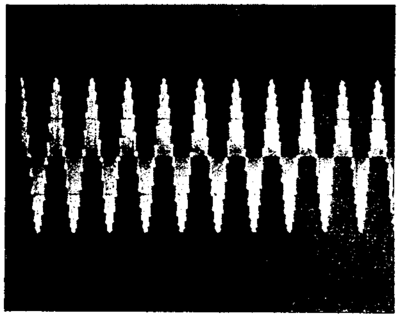


FOURIER TRANSFORM
(POLAR COORDINATES)

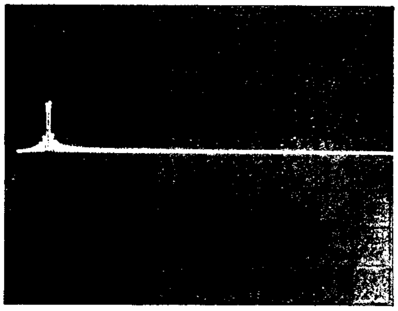


FUNCTION NOT PERIODIC
IN SAMPLING WINDOW

INPUT

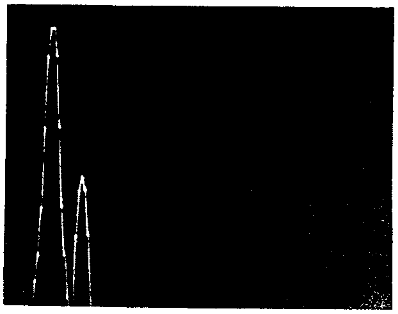


FOURIER TRANSFORM
(POLAR COORDINATES)



The Fourier Analyzer provides a function, called Hanning, to correct this. Looking at the polar display (of function not periodic in window), there is no indication that more than one signal may be present. Yet after Hanning, it is suddenly quite clear that an additional signal of lower amplitude is present, as shown below. Amplitude uncertainty caused by sampling window error or leakage can also be minimized via Hanning.

AFTER HANNING
(LOG DISPLAY)

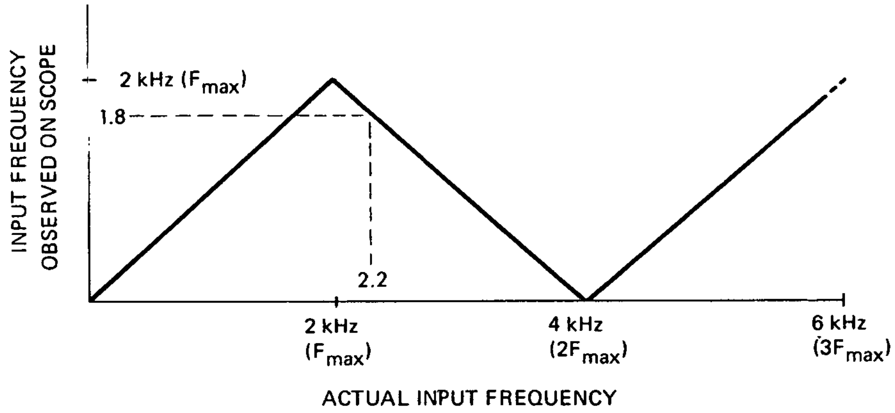


## ALIASING

Aliasing is a problem that develops if you set the ADC's SAMPLE CONTROLS for a certain maximum input frequency (MAX FREQ or $F_{max}$), and then feed in data which contains frequencies that are higher than this setting. The higher frequencies will "fold back", appearing as lower frequencies, within the range of the display -- thus the input will appear to contain frequencies which, in fact, are not there at all.

Refer to the illustration below and you should be able to see how, with a MAX FREQ setting of 2 kHz, a frequency of 2.2 kHz will show up at the "1.8 kHz" position in the display.

Aliasing is not a fault of the Fourier Analyzer itself, but is a direct result of the sampling theorem and is common to all digital signal analyzers. You can avoid aliasing by making sure that the MAX FREQ you set is higher than the highest frequency in the data at the Analyzer's ADC input; if necessary, or desired, a low pass filter can be connected before the input to limit the frequencies it receives.



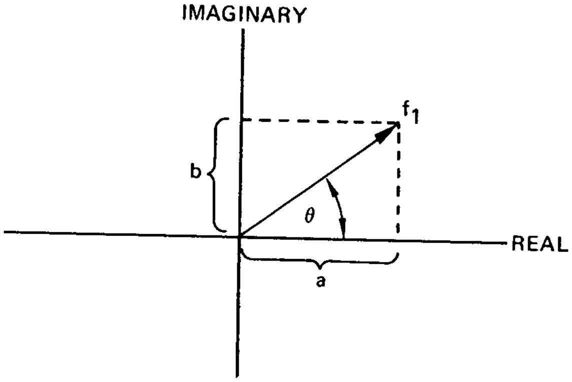
*THUS 2.2 kHz will be seen as 1.8 kHz*
*4.0 kHz will be seen as 0 kHz*
*etc.*

*ERRONEOUS READINGS! To prevent, set $F_{max}$ higher than the highest frequency in the data.*

## CONJUGATE MULTIPLICATION (MULT* KEY)

We can describe a frequency point in the Fourier transform as a + ib, "a" being the amplitude in the real display, "b" the amplitude on the imaginary display, and "i" being $\sqrt{-1}$. In conjugate multiplication, the a + ib of each frequency point is multiplied by the a - ib of the same point.

The result is $a^2 + b^2$. These $a^2 + b^2$ values for all frequency points make up the power spectrum. The conjugate multiply causes the loss of all phase information (angle $\theta$ disappears) because the power spectrum is composed solely of real values (no "i" in $a^2 + b^2$). This can be seen by switching the power spectrum display to imaginary, which should be all 0.



*MULT* (conjugate multiplication):*
*$(a + ib)(a - ib) = a^2 + b^2$*

⇩

*A Point in the Power Spectrum*

## FURTHER DETAILS ON THE N/2 FREQUENCY DOMAIN POINTS

When data in a data block represent the values associated with a spectrum or other function of frequency, they are stored differently than when they represent a time series. A time series of N independent points results in a frequency spectrum of N/2 independent frequencies.

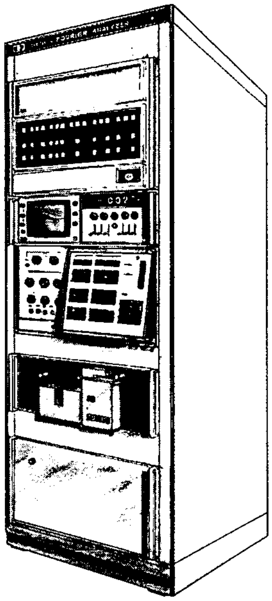
In the Fourier Analyzer, N/2 positive frequencies (plus dc) are computed, stored, and displayed, from an N-point real time series. Each frequency has two independent values -- a "real" (cosine) value, and an "imaginary" (sine) value; the imaginary values for dc and $F_{max}$ are zero, and are not stored. The actual arithmetic is as follows: There are N/2 positive real frequency values, plus the value of dc -- thus (N/2)+1 real points. There are (N/2)-1 imaginary frequency values, since there are no imaginary values for dc or $F_{max}$. Adding the number of real and imaginary points together, we get:

$$(N/2)+1 + (N/2)-1 = N$$

points in the frequency domain from N points in the time domain. We store frequency domain data in a N-word data block as follows: the real value of dc is stored in the first location, the real value of $f_{max}$ is stored in the second location, and the remaining locations are assigned in pairs to the data for the remaining frequencies. The data stored for the remaining frequencies depends on the mode selected; it will be either real and imaginary values, real values only (in double-precision arithmetic), or polar magnitude and phase values.

# SECTION IV

## Fourier Transform Theory: Mathematical Background



## FOURIER SERIES

We know that time functions are often conveniently interpreted by the analysis of their frequency content. This approach is derived from the work of French mathematician Jean Baptiste Fourier. Fourier discovered that periodic time functions can be broken down into an infinite sum of properly-weighted sine and cosine functions of the proper frequencies. The mathematical statement of this discovery is:

$$x(t) = a_o + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \qquad (4\text{-}1)$$

where T is the period of x(t), that is, x(t) = x(t + T)

When the coefficients $a_n$ and $b_n$ are calculated using the equations derived by Fourier, the amplitude of each sine and cosine wave in the series is known. Equivalently, when the coefficients $a_n$ and $b_n$ are known, the magnitude and phase at each frequency in x(t) is determined, where

$$\sqrt{a_n^2 + b_n^2}$$

is the amplitude at the frequency $f_n = (n/T)$, and $\tan^{-1}(b_n/a_n)$ is the corresponding phase.

## THE FOURIER TRANSFORM

The Fourier Series is a useful tool for determining the frequency content of a time-varying signal. However, the Fourier Series always requires a periodic time function. To overcome this shortcoming, Fourier evaluated his series as he let the period of the waveform approach infinity. The function which resulted is known as the Fourier Transform. The Fourier Transform Pair is defined as:

$$S_x(f) = \int_{-\infty}^{\infty} x(t)\ e^{-i2\pi ft}\ dt \qquad \text{(Forward Transform)} \qquad (4\text{-}2)$$

$$x(t) = \int_{-\infty}^{\infty} S_x(f)\ e^{i2\pi ft}\ df \qquad \text{(Inverse Transform)} \qquad (4\text{-}3)$$

Where $e^{\pm 2\pi ft} = \cos(2\pi ft) \pm i \sin(2\pi ft)$, is known as the kernel of the Fourier Transform.

$S_x(f)$ is called the Fourier Transform of x(t). $S_x(f)$ contains the amplitude and phase information at every frequency present in x(t) without demanding that x(t) be periodic.

From the foregoing discussion of Fourier Series and Transform analysis, one sees that both of these techniques may be viewed as mathematical filtering operations.

## THE DISCRETE FINITE TRANSFORM

The Fourier Analyzer utilizes a digital computer to calculate Fourier transforms of time-varying voltage signals. We will examine the results of computing the Fourier transform digitally, considering the forward transform,

$$S_x(f) = \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi ft}\, dt \qquad (4\text{-}4)$$

In order to implement the Fourier transform digitally, one must convert the continuous input signal into a series of discrete data samples. This is accomplished by sampling (measuring) the input waveform, x(t), at certain intervals of time. We will assume that the samples are spaced uniformly in time, separated by an interval $\Delta t$. In order to perform the integral (4-4), the samples must be separated by an infinitesimal amount of time (i. e., $\Delta t \to dt$). Due to physical constraints on the analog-to-digital converter, this is not possible. As a result we must calculate

$$S_x''(f) = \Delta t \sum_{n=-\infty}^{n=+\infty} x(n\Delta t)\, e^{-i2\pi fn\Delta t} \qquad (4\text{-}5)$$

Where $x(n\Delta t)$ are the measured values of the input function.

Equation (4-5) states that, even though we are dealing with a sampled version of x(t), we can still calculate a valid Fourier transform. However, the Fourier transform as calculated by (4-5) no longer contains accurate magnitude and phase information at all of the frequencies contained in $S_x(f)$. Rather, $S_x''(f)$ accurately describes the spectrum of x(t) up to some maximum frequency ($F_{max}$) which is dependent upon the sample spacing, $\Delta t$. The determination of $F_{max}$ is discussed further on page 4-4.

In order to calculate $S_x''(f)$, we must take an infinite number of samples of the input waveform. As each sample must be separated by a finite amount of time, one would have to wait forever for the calculation of $S_x''(f)$ to be completed. Clearly then, we must limit our observation time in order to calculate a useful Fourier transform. Let us assume that the input signal is 'observed' (sampled) from some zero time reference to time T seconds. Then we have

$$\boxed{T/\Delta t = N} \qquad (4\text{-}6)$$

Where N is the number of samples, and T is the "time window".

We see that restricting the observation time to T seconds is equivalent to truncating equation (4-5). As we no longer have an infinite number of time points, we cannot expect to calculate magnitude and phase values at an infinite number of frequencies between zero Hz and $F_{max}$. Equivalently, the truncated version of equation (4-5) does not produce a continuous spectrum. This discrete finite transform (DFT) is given below.

$$S_x'(m\Delta f) = \Delta t \sum_{n=0}^{N-1} x(n\Delta t)\, e^{-i2\pi m\Delta fn\Delta t} \qquad (4\text{-}7)$$

Only Periodic functions have such a 'discrete' frequency spectra. Therefore, equation (4-7) requires that our input function be periodic with period T. Conversely, equation (4-7) assumes that the function observed between zero and T seconds repeats itself with period T for all time. This assumption is made whether or not x(t) is actually periodic. It is apparent that the discrete finite transform, as calculated by (4-7), is actually a sampled Fourier Series.

Note that there are N points in the time series and that, for our purposes, the time series always represents a real-valued function. However, to fully describe a frequency in the spectrum two values must be calculated (i. e., the magnitude and the phase, or the real and imaginary part at the given frequency). As a result, N points in the time domain allow us to define N/2 complex quantities in the frequency domain.

If $F_{max}$ is the maximum frequency present in the spectrum, then

$$\boxed{F_{max} / (N/2) = \Delta f} \qquad (4\text{-}8)$$

where $\Delta f$ is the separation of frequencies (referred to as resolution) in the frequency domain.

## SHANNON'S SAMPLING THEOREM

Shannon states that it requires slightly more than two samples per period to uniquely define a sinusoid. In sampling a time function, this implies that we must sample slightly more than twice per period of the highest frequency we wish to resolve. Translating Shannon's theorem into an equation:

$$\boxed{F_{max} < \frac{1}{2\Delta t}} \qquad (4\text{-}9)$$

For convenience, equation (4-9) will be written:

$$\boxed{F_{max} = \frac{1}{2\Delta t}} \qquad (4\text{-}10)$$

When using equation (4-10) one should remember that the maximum frequency which can be accurately resolved is $F_{max} - \Delta f$.

Substituting (4-10) into (4-6) and employing (4-8) gives:

$$\Delta f = F_{max}/(N/2) = (1/2\Delta t)/(N/2) = 1/N\Delta t = 1/T$$

or

$$\boxed{\Delta f = \frac{1}{T}} \qquad (4\text{-}11)$$

Equation (4-11), as a direct result of Shannon's Sampling Theorem, is a physical law.

## FREQUENCY AMPLITUDE

Let $A_n$ denote the peak amplitude of an input sinusoid of frequency $f_n$. In the Fourier Analyzer, the discrete Fourier transform is implemented such that the amplitude calculated for this $f_n$ is $A_n/2$. Thus, the frequency amplitudes calculated by the Fourier Analyzer must be multiplied by a factor of 2 in order to display peak amplitudes. Similarly, these amplitudes are multiplied by $\sqrt{2}$ in order to display RMS values.

## USEFUL RELATIONS BETWEEN THE TIME AND FREQUENCY DOMAIN

It is appropriate to discuss certain relations between the two domains before we consider the more complex functions.

### Convolution Theorem

given that $$S_x(f) = \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi ft}\, dt$$

and $$S_y(f) = \int_{-\infty}^{\infty} y(t)\, e^{-i2\pi ft}\, dt$$

and $$x(t) * y(t) = \int_{-\infty}^{\infty} x(t)\, y(\psi - t)\, dt$$

where * denotes convolution, the Convolution Theorem states that

$$\int_{-\infty}^{\infty} \left[x(t) * y(t)\right] e^{-i2\pi ft}\, dt = S_x(f) \cdot S_y(f)$$

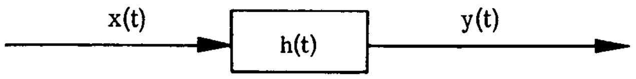
and conversely

$$\int_{-\infty}^{\infty} \left[S_x(f) * S_y(f)\right] e^{+i2\pi ft}\, dt = x(t) \cdot y(t)$$

Simply stated: convolution in one domain corresponds to multiplication in the other domain.

### Convolution, Digital Filtering

Consider the system:



We know that the system output [y(t)] is related to the system input [x(t)] and the system impulse response [h(t)] by

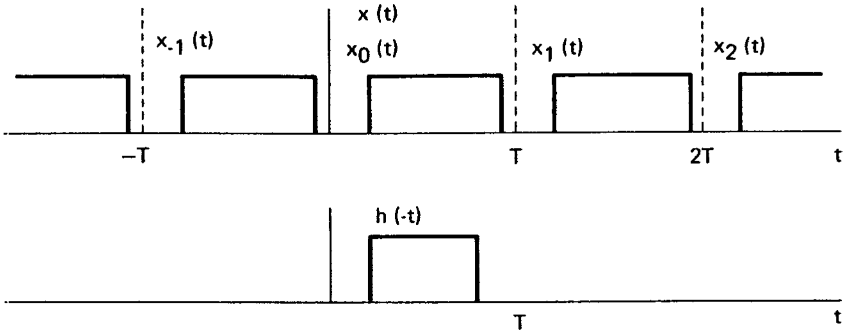
$$y(t) = x(t) * h(t)$$

from the Convolution Theorem, we know that there is an equivalent formulation:

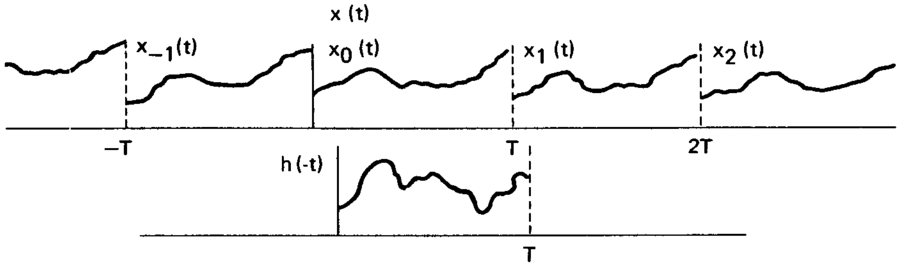
$$S_y(f) = S_x(f) \cdot H(f)$$

This second relation illustrates the manner in which convolution is implemented on the Fourier Analyzer. H(f) can be a (digital) filter transfer function, which may be manually entered via the keyboard. This process of applying a known input to a hypothetical filter and observing the output is very useful.

When we try to perform convolution in the Fourier Analyzer (via the DFT), the Analyzer thinks that both x(t) and h(t) are periodic functions of period T, as shown below (we will consider only the periodicity of x(t) for convenience).



As we envision the shifting of h(-t) along the t axis, it is obvious that the value of the convolution integral is affected dramatically by the periodicity of x(t). Next we examine, in more detail, the nature of the errors introduced by the implied periodicity of x(t). In the Fourier Analyzer, the convolution is effectively computed with the limits (0, T), rather than $(-\infty, +\infty)$. This is to say that we shift h(-t) by an amount T along the positive t axis. If, in this shifting process, there are values of $\psi$ and t such that $h(\psi-t)$, $x_0(t)$ and $x_1(t)$ 'overlap' and are all non-zero, error will be introduced into our calculations. This error is known as WRAP-AROUND ERROR. From the proceeding, it is clear that the amount of wrap-around error introduced is dependent upon the record lengths of the waveforms in question. The worst case, and the most common case, occurs when the non-zero record lengths of both x(t) and h(t) are equal to T. In this worst case our convolution is completely distorted by wrap-around errors (see below).

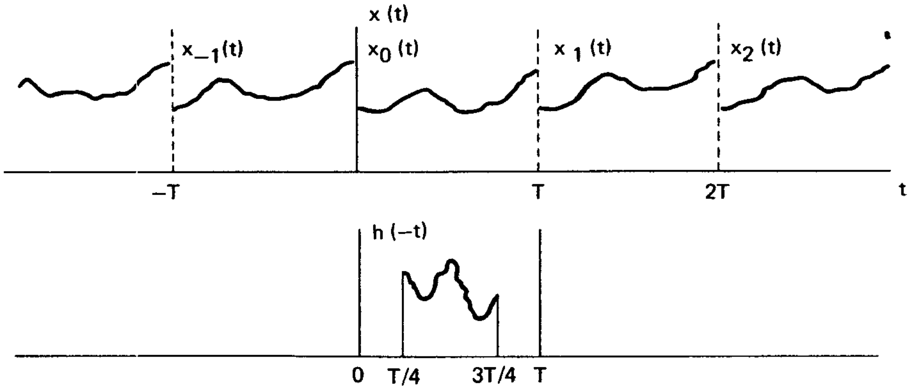


We see that any shift of h(-t), no matter how small, will cause h(-t) to 'overlap' simultaneously with $x_0(t)$ and with $x_1(t)$. In order to help rescue ourselves from this situation, we do the following before we convolve:

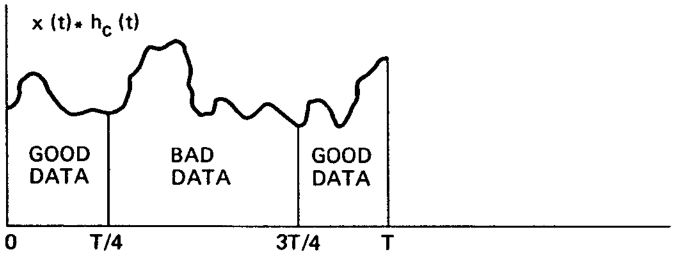
1. If we are performing auto-convolution (i.e., the convolution of a waveform with itself) we must store the waveform in two locations.

2. We clear out each end of <u>one</u> of the waveforms by an amount T/4 (in cross-convolution, either record may be cleared in this manner; in auto-convolution, either of the two blocks where the waveform is stored is cleared as indicated above).

This process is illustrated below--where $h_c(-t)$ is the cleared function.



Now, as we shift h(-t), we see that the first shift of T/4 provides us with valid data (no overlap). The next T/2 of shift provides us with invalid convolution due to the overlap of $h_c$, $x_0$, and $x_1$. The final T/4 of shift again provides us with good convolution data. The resulting convolution is valid 'on the ends', and invalid in the middle due to wrap-around error, as shown below.



We see that, by forfeiting one half of one of our waveforms we gain T/2 of valid convolution data.

This T/2 seconds of valid convolution data may be viewed as N/2 lags, where each lag is of length $\Delta t$. In practice, the bad data (T/2 worth in the middle) is usually cleared out so that only valid information is displayed.

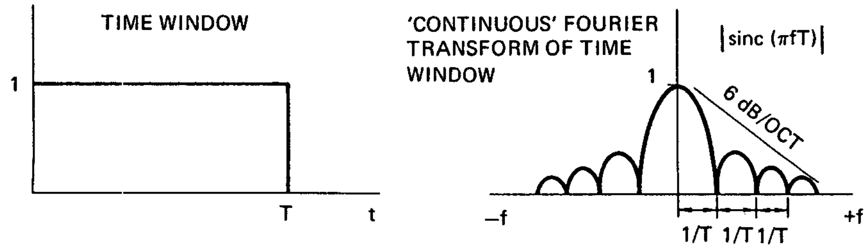
## The Auto Power Spectrum

The auto-power spectrum of a function x(t) is defined as:

$$G_{xx} = S_x(f) \cdot S_x{}^*(f)$$

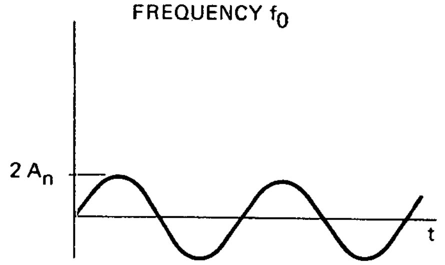
All of the frequency components of $G_{xx}$ are purely real and positive.

If $A_n$ represents the Peak Amplitude of an input sinusoid of frequency $f_n$, then the Fourier Analyzer displays $A_n{}^2/4$.

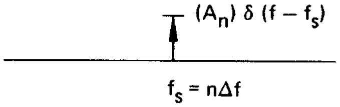


It is now appropriate to view the frequency domain effects of the implied periodicity of the DFT. As we are concerned with power spectral quantities, the magnitude only of the transformed time window will be considered. The slopes indicate side lobe "roll-off" if plotted in log vs log format.

INPUT SINUSOID OF FREQUENCY $f_0$



CONTINUOUS FOURIER TRANSFORM OF INPUT



For simplicity, we will consider a single sine wave input, although this analysis can be generalized to any input (because Fourier stated that time functions could be represented by a sum of properly-weighted sinusoids).

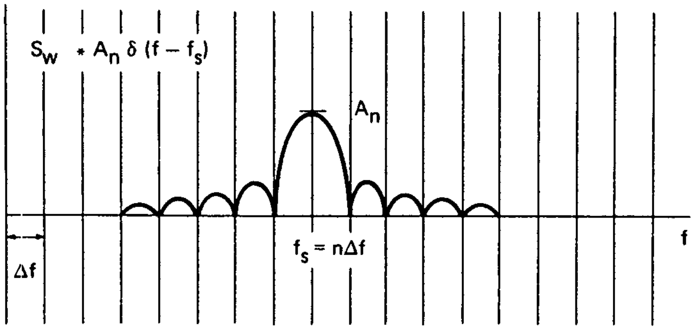
When we take a look at a "sine wave" input to the Fourier Analyzer, we are, in effect, observing a function that is a sine wave which exists for all time multiplied by the Analyzer's time window of duration T (see curves above). However, we know that multiplication in the time domain corresponds to convolution in the frequency domain (the convolution theorem). As we are investigating the frequency domain effects of the DFT, it is convenient to interpret the effects of the time window multiplication as a convolution of the Fourier transform of the time window with the Fourier transform of the true sine wave. These Fourier transforms are shown above. It turns out that convolution of any function with a delta function is a rather trivial operation. Specifically,

$$F(f) * \delta(f - f_0) = F(f - f_0)$$

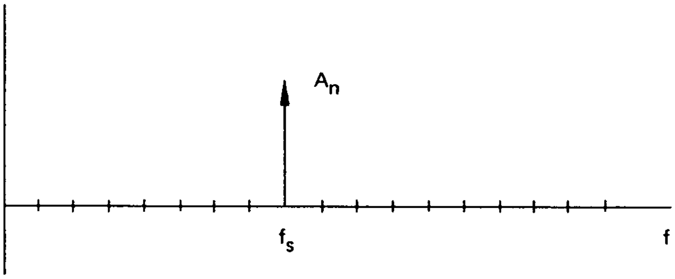
In other words, when a function, F(f), is convolved with a delta function which is non-zero at $f_0$, the resultant of the convolution is the original function shifted by an amount $f_0$. Considering the specific case of $S_w(f) * \delta(f - f_s)$, where $S_w$ is the Fourier transform of the window function, the resultant is $S_w(f - f_s)$. The window transform is now centered about $f_s$. It is important to note that this convolution effectively takes place simultaneously with the multiplication of the time window and the sinusoid. Even though we cannot observe the results of this convolution until the DFT takes us into the frequency domain, the convolution took place before the DFT. When the DFT shows us the frequency domain we can only observe the results of this convolution at $\Delta f$ increments. It is as if we are looking through a piece of paper with fine slits separated by a distance $\Delta f$.

There are two basic situations to consider:

1. The sine wave has an integral number of periods in the window. That is, if $T_s$ is the period of the sine wave and n is an integer, then $T = nT_s$. We know that $1/T_s$ is $f_s$, the frequency of the sine wave, and $1/T = \Delta f$. Substituting these values into the expression $T = nT_s$ gives $f_s = n\Delta f$. This means that, if the sinusoid has an integral number of periods in the time window, its Fourier transform will lie exactly on one of the lines in the frequency domain which we are capable of observing. The following illustration indicates the result of the convolution of $|S_w(f)|$ with an input sine wave of amplitude $2A_n$ and frequency $n\Delta f$--where n is an integer.

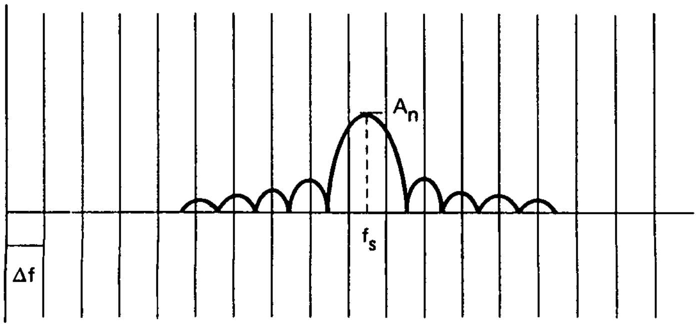


Now when we try to observe this result by taking the DFT we are only allowed to view the convolution at Δf increments. As a result, we see

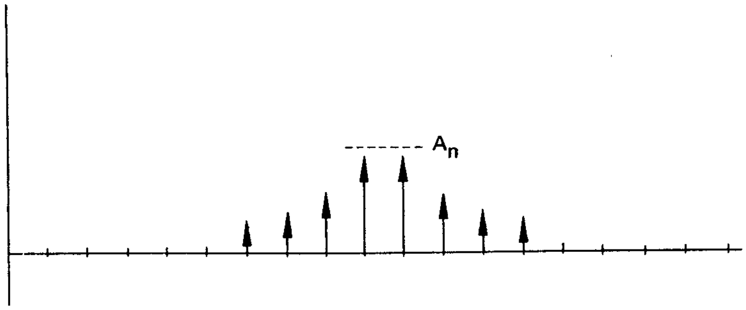


Due to the location of the nulls of $|S_W|$, it turns out that the DFT of the function which is the sine wave modified by the time window is the same as the Fourier transform of the unmodified sine wave. As the DFT assumes that what it sees repeats itself for all time, and as we have demanded that this sine wave be periodic in the window, we should not be surprised that the DFT and the Continuous Fourier transform are equivalent for this case.

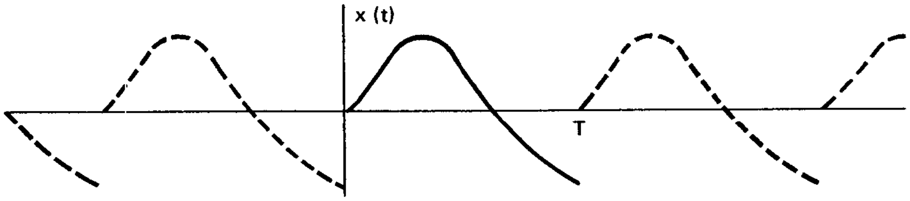
2. If the input sinusoid does not have an integral number of periods in the time window, the equation $f_S = n\Delta f$ is still valid, however n is no longer allowed to be an integer. Let us view the 'continuous convolution' in this case.



When we take the DFT of this function we see the following



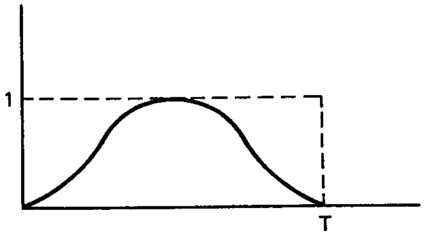
We see that the DFT of an input sinusoid which does not have an integral number of periods in the time domain will appear at more than one frequency (although the actual location of $f_s$ is uncertain by less than Δf. The amplitude of the sine wave will be reduced from its true value. This all occurs because the DFT thinks it is operating on a function that looks like this:



which is, clearly, not a sine wave.

We cannot eliminate these problems entirely, however we can reduce many of the leakage effects, and gain accurate amplitude information at the expense of less precise frequency resolution. We accomplish this trade-off by using the Hanning window.

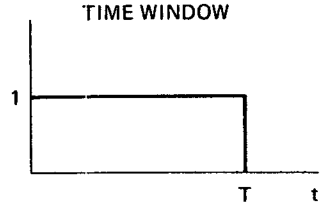
The Hanning window modifies the effective shape of the time window by multiplying the window by the function $|1/2 - 1/2 \cos (2\pi t/T)|$ . The effective window then takes on the shape of this Hanning function. The new window looks like:



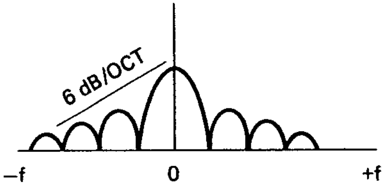
Note that this window eliminates discontinuities at the ends of the time record.

To implement this Hanning window in the Fourier Analyzer, we multiply the Hanning window and the gated input wave. This multiplication of two time domain functions results in convolution in the frequency domain. The convolution of the two time domain window functions is illustrated below.

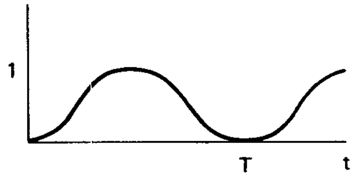
TIME WINDOW



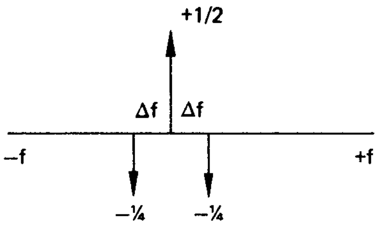
WINDOW TRANSFORM



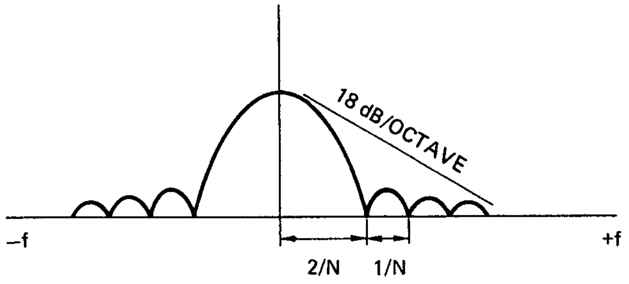
HANNING WINDOW



HANNING WINDOW TRANSFORM

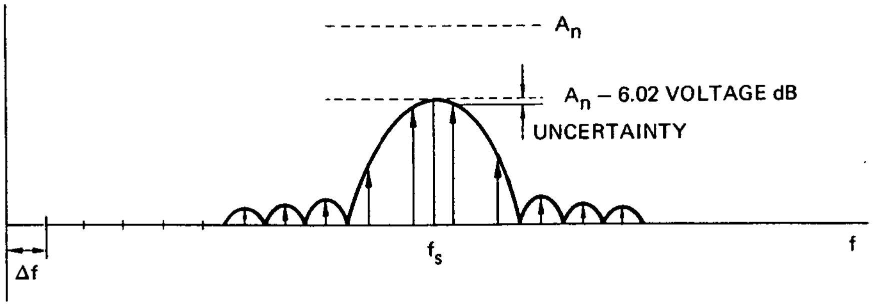


The result of the convolution of the two window functions is



We see that the main lobe of our modified window function is widened and flattened, while the side lobes are greatly reduced in amplitude.

If we now view the convolution of this modified window transform and a sine wave which is not periodic in the window, we have



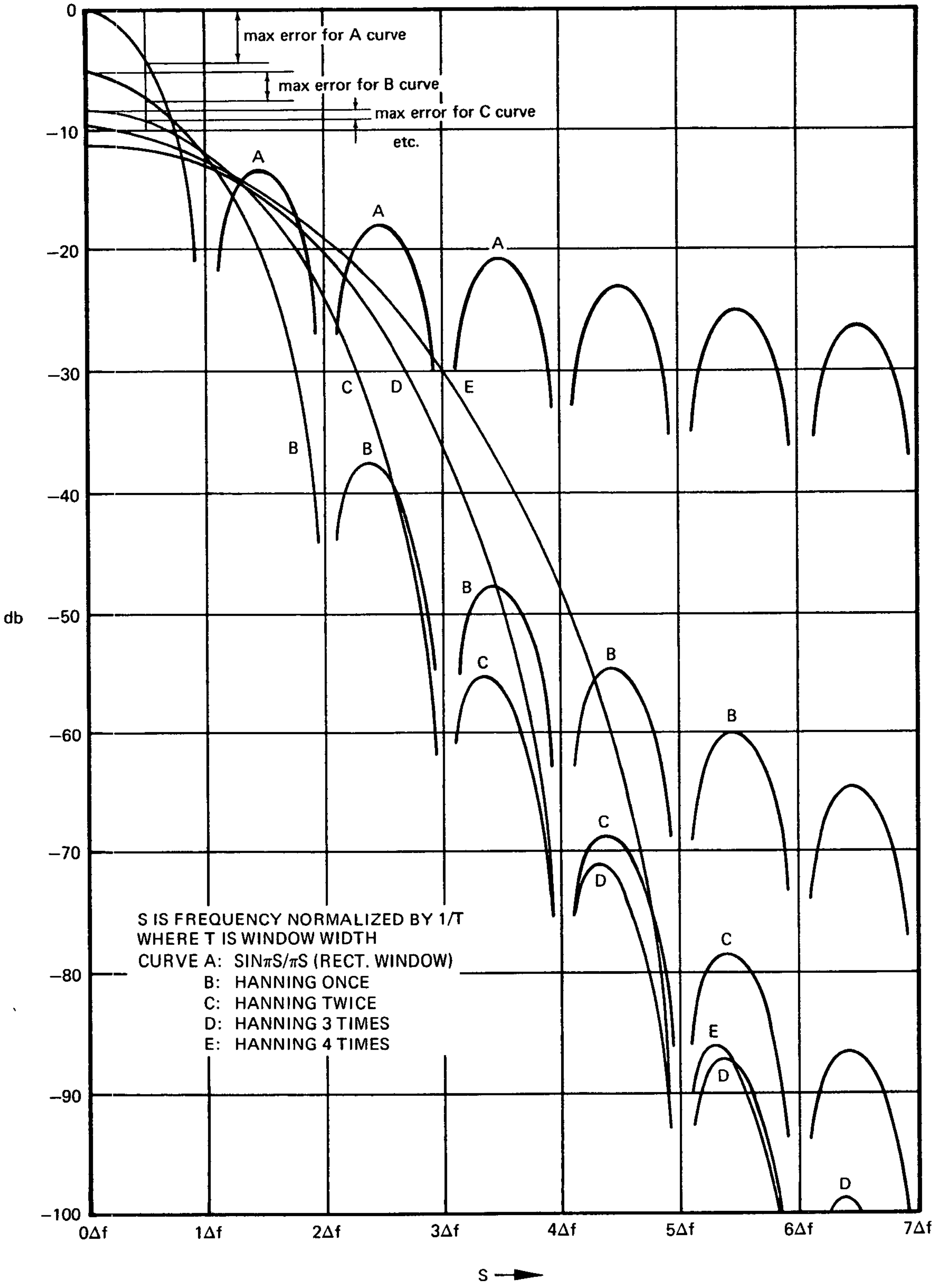
We see that, while one Hanning attenuates the amplitude by 6.02 dB, the uncertainty (unknown amplitude error) of the measurement is greatly reduced due to the relative flatness of the main lobe of the modified window function. It is also clear that the price we pay for reducing the amplitude uncertainty is degradation of frequency resolution (exactly where the frequence $f_s$ is located is now less certain).

One can Hann repeatedly. Each time the window is Hanned, the side lobes of its transform are attenuated by 12 dB/octave (40 dB/decade), while the main lobe of the transform is widened by 2Δf. Each Hanning also has a corresponding attenuation factor which must be added to the results (or which the results must be multiplied by, if the dB's are converted to their equivalent power ratio) in order to display true amplitude. We have seen that for one Hanning, this factor is 6.02 dB (a multiplication factor of 2). In the case of two, three, and four Hannings the attenuation factors are 9.52 dB (2.67), 10.10 dB (3.2) and 11.26 (3.66), respectively. (These are total attenuation factors for Hanning two, three, and four times, not the incremental attenuation factor for each Hanning. Thus, for example, after three Hannings, one would add 10.10 dB to each of the main lobes (or multiply each lobe by 3.2). The uncertainty then between the resulting values and the correct amplitude is as shown in Figure 4-1, curve D. As worst case amplitude error occurs when the sine wave is precisely in the middle of two channels, it is seen that four Hannings reduce this error to less than 1/2 dB.

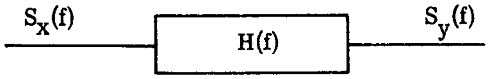
So far we have only taken advantage of the fact that Hanning flattens the main lobe of the window function. When the spectrum of interest contains two or more sine waves, the increased side lobe rolls off, which is a result of Hanning, enhances the Analyzer's ability to separate these frequencies.

Taking the foregoing into account, Auto-Power Spectral analysis may be used to determine certain characteristics of a system transfer function.

FIGURE 4-1. HANNING LINE SHAPES



Consider the system:



$$S_y(f) = S_x(f) \cdot H(f)$$

$$G_{yy} = (S_xH)(S_xH)^* = (S_xH)(S_x^* \, H^*) = |H(f)|^2 \, G_{xx}$$

or

$$|H(f)|^2 = G_{yy}(f)/G_{xx}(f)$$

We see that Auto Spectral Analysis can provide us with the magnitude characteristics of the transfer function. No phase information is present. A discussion of why we might choose this approach is included in the section on Ensemble Averaging, beginning page 4-17.

## Voltage Spectrum

The "voltage spectrum" is defined as $|S_X(f)|$. The voltage spectrum is the positive square root of the Auto Power Spectrum. The leakage problems we have discussed with respect to Auto Spectrum also exist in Voltage Spectrum and are dealt with in exactly the same manner as in the auto spectral case.

## Auto-Correlation (Auto-Covariance) Functions

Auto-Correlation $R_{XX}$ is defined as the inverse Fourier transform of the Auto Power Spectrum:

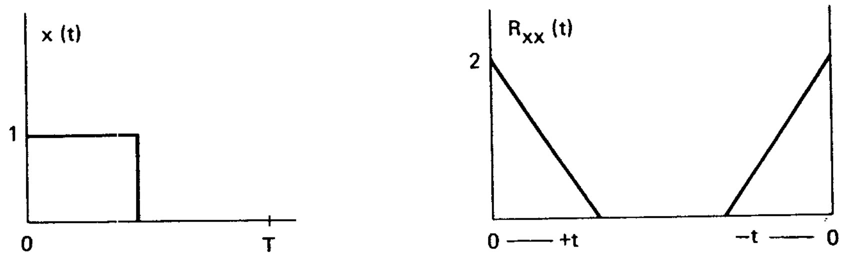
$$R_{xx}(t) = \int_{-\infty}^{\infty} |S_x(f)|^2 \, e^{i2\pi ft} \, df = \int_{-\infty}^{\infty} S_x(f) \, S_x{}^*(f) \, e^{i2\pi ft} \, df$$

Since we are only considering real input functions

$$\boxed{R_{xx}(t) = \int_{-\infty}^{\infty} x(\psi) \, x(\psi - t) \, d\psi}$$

Some authors refer to $R_{XX}(t)$, defined above, as the Auto-Covariance function of x(t). They define the Auto Correlation function as $\rho_{XX} = R_{XX}(t)/R_{XX}(0)$.

The correlation integral is much like the convolution integral, only the function $x(\psi)$ is not 'flipped' before it is shifted past itself as is the case in convolution. As a result, the auto-correlation of a pulse is displayed as shown below:



As one would expect, wrap-around error manifests itself in correlation just as it did in convolution. To overcome the effects of wrap-around error, we clear T/4 off of the ends of one of the records (as before) and correlate (by calculating the Auto Spectrum and inverse transforming). Next we clear out the invalid portion of the correlation, much as we did with convolution. The valid correlation data exists in T/4 wide bands at the ends of the record.

### The Cross-Power Spectrum

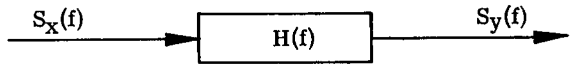
The cross-power spectrum of two signals is defined as

$$G_{yx}(f) = S_y(f) \cdot S_x{}^*(f)$$

Note that $G_{yx}$ in general assumes both positive and negative values. The relative phase between the signals is preserved in cross-spectral analysis. The implied periodicity of the DFT gives rise to leakage effects similar to those we have seen in auto-spectral analysis. The leakage is diminished and the amplitude information is enhanced by Hanning just as was done in the auto-spectral case.

A zero value of cross-spectrum indicates that one or both of the individual spectra are zero at that frequency. A relatively large value for the cross-spectrum likewise indicates that both of the individual spectra have large values at that frequency. The cross-spectrum indicates the relationship between two signals.

If x(t) represents the input to a system, and y(t) represents the system output, cross-spectral analysis can be used to determine the system transfer function (as shown below).



$$S_y(f) = S_x(f)\ H(f) = > G_{yx}(f) = \left[S_x(f)\ h(f)\right] S_x{}^*(f)$$

or

$$\boxed{H(f) = G_{yx}(f)/G_{xx}(f)}$$

Thus, using cross-spectral analysis we are able to describe both the magnitude and the phase of the transfer function.

When we are doing cross-spectrum of system outputs and inputs (as is usually the case), the relationship between input and output is not sufficient information to determine an accurate transfer function. In addition to relationship (i. e., relative magnitude and phase of the two signals) we must know whether or not the system output was totally caused by the system input.

That is, noise and/or non-linear distortion can cause large outputs, and therefore large values for the cross-spectrum, at various frequencies, and thus introduce errors in calculating the transfer function.

In order to determine 'causality' we examine the coherence function, $\gamma^2$

$$\gamma^2 \triangleq \frac{|G_{yx}(f)|^2}{G_{xx}(f)\, G_{yy}(f)} \qquad 0 \leq \gamma^2 \leq 1$$

If the coherence is 1, our system has perfect causality. Low coherence at given frequencies indicates that our transfer function has inaccuracies at those frequencies.

## The Cross Correlation (Cross-Covariance) Function

Cross-correlation is defined as the inverse Fourier transform of the cross-power spectrum.

$$R_{yx}(t) = \int_{-\infty}^{+\infty} x(\psi)\, y(\psi - t)\, d\psi$$

Again, some authors refer to this as the cross-covariance function. They define cross-correlation as:

$$\rho_{yx}(t) = \frac{R_{yx}(t)}{\sqrt{R_{xx}(0)\, R_{yy}(0)}} \qquad -1 \leq \rho_{yx}(t) \leq 1$$

Even though the correlation function and the power spectrum theoretically have equal information content, more samples per period may be required to effectively interpret the time domain (correlation) function.

Only if $\rho_{xy}$ is calculated, can results of independent experiments be compared.

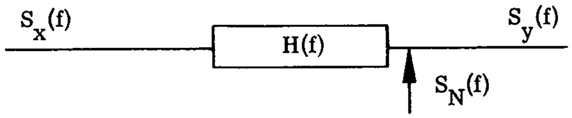
Wrap-around error is present in cross-correlation measurements. It is dealt with in the identical manner as auto-correlation.

## Ensemble Averaging

In the real world, signals are somewhat ill-defined due to the presence of noise. Oftentimes signals of interest are completely obscured by this phenomenon. Noise, however, is random and independent of the signals of interest. If we take repeated time or frequency records of noisy signals, and average these records, the noise will average to zero (if enough averages are taken) and the previously hidden signal will become visible. It is because of the above that the Fourier Analyzer was designed with the capability of storing averaging-loop programs.

It is instructive to observe some of the behavior of the "useful functions" we have been discussing in the presence of noise.

The auto-spectra and transfer function magnitude.



ideally:

$$S_y(f) = S_x(f) \cdot H(f) \Rightarrow G_{yy}(f) = G_{xx}(f)\,|H(f)|^2$$

with noise

$$S_y(f) = S_x(f)\,H(f) + S_N(f)$$

The result of the presence of noise on $G_{yy}(f)$ is:

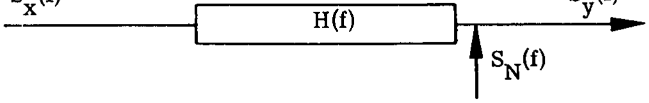
$$\begin{aligned} G_{yy} &= (S_x \cdot H + S_N)(S_x \cdot H + S_N)^* \\ &= (S_x \cdot H + S_N)(S_x^* H^* + S_N^*) \\ &= G_{xx}\,|H|^2 + S_x H S_N^* + S_x^* H^* S_N + |S_N|^2 \end{aligned}$$

$S_x S_N^*$ and $S_x^* S_N$ approach zero as the number of averages increase, and may be neglected. Note that $|S_N|^2 > 0$ for each record, and will not average to zero but will remain as an error term.

$$G_{yy} = |H|^2 G_{xx} + |S_N|^2 \Rightarrow \boxed{|H_n|^2 = \frac{G_{yy}}{G_{xx}} - \frac{|S_N|^2}{G_{xx}}}$$

We see that averaging will not eliminate the errors due to noise in calculating the transfer function magnitude by the auto-spectral method.

Cross-spectra and transfer function calculations:



ideally:

$$G_{yx} = S_y S_x^*$$

$$H(f) = G_{yx}/G_{xx}$$

with noise:

$$G_{yx} = (S_x H + S_N)\, S_x^* = HG_{xx} + S_N S_x^*$$

Note that the $S_N S_x^*$ term becomes very small as the number of averages increases

$$\boxed{H_n(f) = \frac{G_{yx}}{G_{xx}} = H + \frac{S_N S_x^*}{G_{xx}}}$$

indicates that the effect of $S_N S_x^*$ is even less on the transfer function ($G_{xx} > 0$).

Even though cross-spectral analysis greatly reduces the error due to noise, error is still present. How much, is indicated by the coherence function.

## Coherence Function

Let's examine the coherence function further, disregarding noise.

By definition:

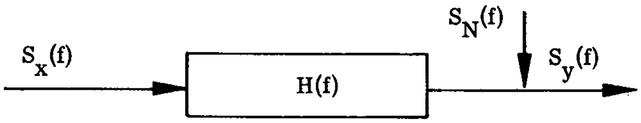
$$\gamma^2 = \frac{|G_{yx}|^2}{G_{xx}G_{yy}} = \frac{G_{yx} \cdot G_{xy}}{G_{xx}G_{yy}}$$

$$= \frac{(S_y S_x^*)(S_x S_y^*)}{(G_{xx})(G_{yy})}$$

$$= \frac{G_{xx}G_{yy}}{G_{xx}G_{yy}} \equiv 1$$

The coherence function is independent of the transfer function. This function may be viewed as a mathematical ploy, constructed to detect the presence of noise (or non-linear distortion). If a two-port system is linear and noise-free, the input to output coherence must be 1.

Now, let's examine the coherence function for this system in the presence of noise.



$$S_y = H\,S_x + S_N$$

$$G_{yy} = |H|^2 G_{xx} + H\,S_x S_N^* + H^* S_N S^*_x + |S_N|^2$$

$$G_{yx} = H\,G_{xx} + S_N S_x^*$$

As we average, the cross-terms, ($S_xS_N^*$ and $S_NS_x^*$) approach zero, assuming that signal n and noise N are not related. Then the expression for the coherence becomes:

$$\delta^2 = \frac{|H \cdot \overline{G_{xx}}|^2}{\overline{G_{xx}}\,(|H|^2\,\overline{G_{xx}} + |\overline{S_N}|^2)}$$

$$\delta^2 = \frac{|H|^2\,\overline{G_{xx}}}{|H|^2\,\overline{G_{xx}} + |\overline{S_N}|^2} < 1$$

The smaller the value of the coherence function, the less the input to output 'causality' in a system. This function is a measure of the noise-error present in the transfer function at specific frequencies.

In concluding the topic of ensemble averaging, it is relevant to discuss the relationships between time domain and frequency domain averaging.

As time domain and frequency domain representations of a signal contain equal information, we expect time domain averaging and frequency domain averaging to be equivalent processes. Practically, this is not entirely true. A sync pulse if required to compute a meaningful average of a time function. This sync (trigger) signal preserves the relative phase of the samples. A sync is also required to average Fourier transforms, but power spectrum averaging requires no such trigger.

Thus, phase information can be obtained from the averaged cross-power spectrum without prior knowledge of the signals being measured.

# INDEX

# INDEX (Cont'd)

## INDEX (Cont'd)

HEWLETT PACKARD

PRINTED IN USA