# MEASUREMENT AND ENHANCEMENT OF WAVEFORM DIGITIZER PERFORMANCE

Lyle Ochs
Tektronix, Inc., Beaverton, Oregon 97077

## SUMMARY

Both measurement and enhancement of waveform digitizer performance traditionally have been difficult problems. Current trends to digitizing systems that incorporate minicomputers (or specialized microprocessors) lead to systems capable of new approaches and more complete solutions to this class of problems.

This paper presents a summary of some current efforts at Tektronix in these areas with primary emphasis on measuring dynamic performance of waveform digitizers and on the correction of CRT distortion in the R7912 Transient Digitizer.

## INTRODUCTION

The scientist or engineer attempting to solve a particular digital signal processing problem typically faces a difficult task in selecting the digitizing system that best fits his needs. The specifications, historically, have been of the dc or static type and are often unclear or incomplete.[1,2]

We are both concerned with and affected by the lack of adequate dynamic specifications for waveform digitizers. In addressing this question over the past few years, we have developed a computer-oriented approach requiring very little specialized instrumentation. It is applicable to most A/D converters or waveform digitizers. While certainly not replacing the more specialized systems and component testing necessary in A/D converter design, this computer-oriented "black box" approach does provide a solid foundation for evaluating overall digitizer performance.

Addressing questions of performance specifications leads directly to questions of enhancing performance. Some of these are typically: "Can the digitizer be easily calibrated to improve dc gain accuracy?," "Can its bandwidth be enhanced?," and "How can system distortion or nonlinearities be reduced?" The first question can be straightforwardly addressed by systems which include a precision voltage source, ideally a programmable one. Responses to the questions of bandwidth and system distortion are inherently more complex, but on today's minicomputer-based systems, such problems admit more direct and automated solutions.

## MEASURING DIGITIZER PERFORMANCE

### The Test Setup

In the past, test setups for measuring digitizer performance have involved specialized analog instrumentation and reference signals. One such configuration consists of a reference signal, A/D, D/A and comparator as diagrammed in Fig. 1. The comparator output defines an analog error signal which is examined with an oscilloscope or spectrum analyzer. Valid performance specifications result only if the D/A and comparator contributions to the analog error signal are small with respect to those of the A/D. This can be an especially troublesome problem in testing the higher resolution, higher speed A/D converters available today.

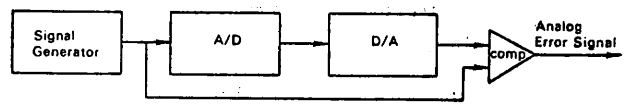



Fig. 1. An analog setup for testing A/D performance.

A more direct procedure is to interface the digitizer directly to a computer and analyze the output in its own domain -- digitally. A block diagram for this test setup includes a signal generator, digitizer, minicomputer, graphics terminal, and a set of analysis routines, as in Fig. 2. Typical choices for the signal generators include precision voltage sources for static testing and sine wave generators or random noise generators for dynamic tests.

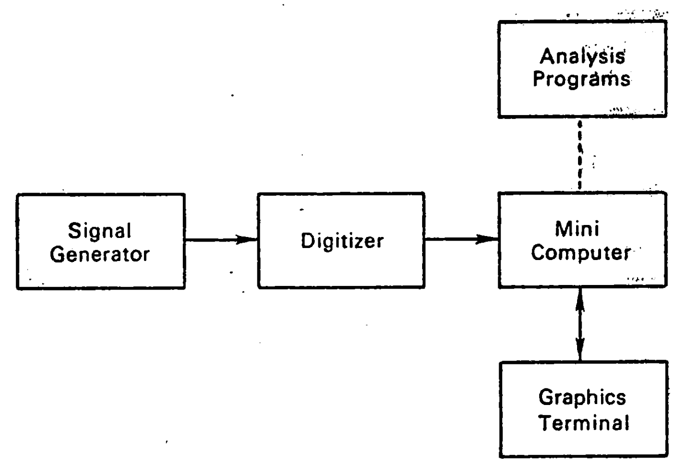



Fig. 2. Direct digital setup for testing digitizer performance.

In this digital approach, there are no extraneous error components (from the D/A and comparator), nor are there problems synchronizing comparator input as in the analog setup. On the other hand, the analog approach uses the analog input signal as a reference, while the reference signal is only inferred in the direct setup and the subsequent analysis. Thus, specifications for the reference signal are more critical, and they must exceed theoretical accuracy, resolution, and noise limits of the digitizer to be tested. For sine wave testing, the insertion of a bandpass filter between the signal generator and digitizer substantially reduces these problems by rejecting harmonic distortion components and reducing noise bandwidth.

The minicomputer in the test setup of this study was a PDP-11, and the analysis routines were written in FORTRAN. Because the results of the computer analyses were frequently presented in graphical form, the graphics terminal was an important component of the test setup.

Dynamic Testing with Sine Waves

Focusing on dynamic testing, both the type of reference signals and the computer analysis approach need to be determined. A sine wave is the obvious choice for the reference signal. It is relatively easy to generate nearly perfect high frequency sine waves and their properties are well understood. The actual computer analysis approach is not so obvious. Efforts reported elsewhere[5,6] have relied primarily on a frequency domain approach to determine digitizer performance, although a time domain approach is also possible. This alternative method involves synthesizing the analog input signal via computer analysis of digitizer output, and allows for both time and frequency domain analyses. The technique is diagrammed in Fig. 3.

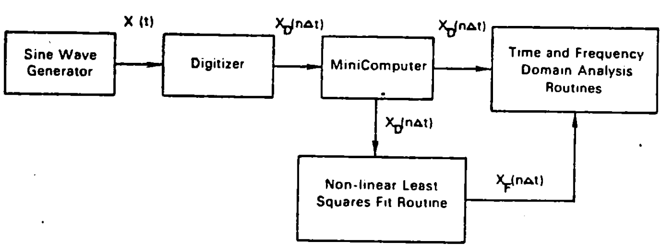



Fig. 3. Analysis approach indicating signal flow and major program components. X(t) is the analog input signal, $X_D(n\Delta t)$ the sampled and quantized output of the digitizer, and $X_F(n\Delta t)$ the perfectly sampled version of X(t) synthesized in the computer.

In the time domain approach, the analog input signal is a sine wave represented mathematically by the equation

$$X(t) = a \sin(2\pi ft + \phi) + b, \qquad (1)$$

where a, f, $\phi$, and b are the amplitude, frequency, phase and dc offset parameters. The digitizer output is $X_D(n\Delta t)$, $\Delta t$ the sampling interval, and n the sample index. The digitizer output and the frequency of the sine wave reference are fed to a routine that forms initial estimates of the remaining parameters a, $\phi$, and b. All this data is then routed to a nonlinear least squares[7,8,9] routine which determines fitted parameter values. The model with fitted parameters is

$$X_F(t) = \hat{a} \sin(2\pi\hat{f}t + \hat{\phi}) + \hat{b}. \qquad (2)$$

This is taken to be the analog input to the digitizer. $X_F(t)$ can now be "perfectly sampled" in the computer, producing $X_F(n\Delta t)$, and thus the digitizer output can be quantitatively compared with the perfectly sampled analog input, sample by sample. The data $X_F(n\Delta t)$ also can be perfectly quantized -- call this data $X_{FD}(n\Delta t)$ -- and the actual digitizer output $X_D(n\Delta t)$ examined with respect to $X_{FD}(n\Delta t)$, the "output" of a perfect digitizer of the same number of bits. In addition, the fitted model allows derivatives of the input signal to be computed. These are necessary in analyzing timing jitter as described below.

To confirm that this non-linear least squares approach is a viable one, it was tested on computer simulation data. This simulated digitizer output contained both systematic and random error components in varying amounts. Very satisfactory results have been obtained using 256, 512 and 1024 sample points. Sampling rates in these experiments varied from 4/5 to 1/10 of the Nyquist sampling rate for the sine wave frequency in question.

Another point to note is that this approach fits data output by a digitizer (or a simulation program) and so cannot identify digitizer gain, offset, or timing compression/expansion errors. The first two errors can be independently examined using precision voltage sources, the last by comparing the fitted value $\hat{f}$ with the frequency of the sine wave input as measured by a digital counter.

Some Typical Measurement Results

Having fit the model parameters to either computer simulation or actual digitizer data, a variety of analyses are possible. The analysis can be done either in the time or frequency domain, depending on which is more appropriate. By systematically varying the frequency of the input signal and repeating the same analysis for each frequency, dynamic performance versus frequency can be graphically displayed. Some typical results are given below. The data analyzed in all cases is computer generated and corresponds to an 8-bit digitizer.

Effective bits versus frequency is a very convenient measure of dynamic performance. The analog error signal is

$$r(n\Delta t) = X_D(n\Delta t) - X_F(n\Delta t), \qquad (3)$$

and the analog error signal for an ideal digitizer is

$$R(n\Delta t) = X_{FD}(n\Delta t) - X_F(n\Delta t). \quad (4)$$

Thus, the ratio rms[r]/rms[R] provides a measure of digitizer performance relative to an ideal standard. Taking the base two logarithm gives "lost bits." For a B bit digitizer, the effective bits for the digitized data $X_D$ is then

$$EB[X_D] = B - \log_2\left(\frac{rms[r]}{rms[R]}\right). \quad (5)$$

Note that effective bits may decrease due to either systematic or random errors in the digitization process since either type of error will tend to inflate rms[r]. A plot of effective bits versus frequency for the 8-bit simulation data appears in Fig. 4. Five simulated digitizer outputs with input frequencies of 1, 10, 20, 40 and 80 megahertz were examined.

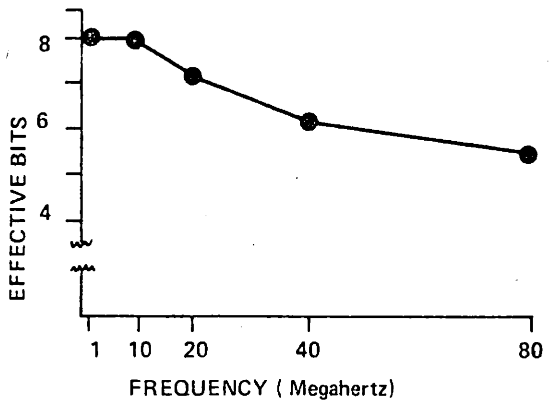



Fig. 4. Effective bits for five sets of simulated digitizer output.

The difference between actual digitizer output and the output of an ideal digitizer defines the encoding errors. Using the above notation, these are given by

$$e(n\Delta t) = X_D(n\Delta t) - X_{FD}(n\Delta t). \quad (6)$$

Encoding errors, as the analog errors defined above, can be graphed directly or can be summarized by a few statistics such as mean, maximum, minimum, and rms. For the 8-bit simulation data examined in Fig. 4, worst case and rms encoding errors were computed. The results appear as Fig. 5.

In making rise-time or level crossing measurements from digitized data, it is important that errors introduced by the digitizing system be known. Using the time domain method of Fig. 3, direct approaches to these problems are possible. For example, amplitude uncertainty at sine wave zero crossings is given directly by the encoding errors $e(n\Delta t) = X_D(n\Delta t) - X_{FD}(n\Delta t)$, at such zero crossings. Since the zero crossings are computable from the fitted model, worst case and rms errors at or near zero crossings can be developed.

Timing jitter, i.e., uncertainty in the exact location of the sampling times, has always

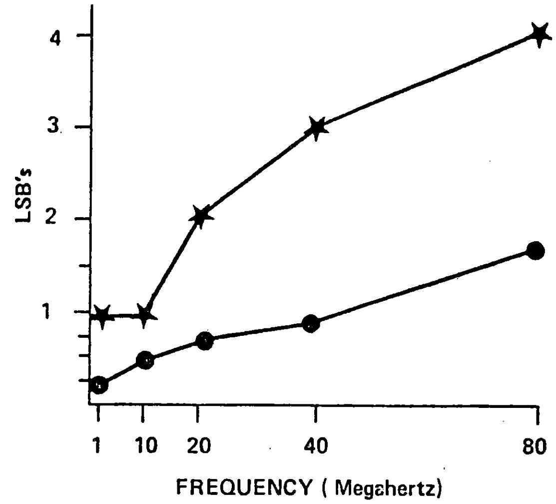



Fig. 5. Worst case (*) and rms (·) encoding errors for the data sets of Fig. 4.

been one of the more difficult performance parameters to measure. But time jitter and amplitude uncertainty at sine wave zero crossings are related by

$$\Delta X = a 2\pi f \Delta T, \quad (7)$$

where $\Delta X$ is the amplitude variation and $\Delta T$ is the time jitter. Reversing this argument, and assuming that the most significant contributor to amplitude variations at zero crossings is time jitter, $\Delta T$ can be estimated as

$$\Delta T = \frac{\hat{a} 2\pi \hat{f}}{\Delta X}. \quad (8)$$

Amplitude variations at zero crossings are available (described above), so jitter values and their statistics can be derived. This measurement typically would be done at an input signal frequency toward the bandwidth limit of the digitizer being tested, since the higher frequencies are more affected by, and therefore better indicators of jitter. The measurement can be repeated a number of times, digitizing a new record each time, to assess stability of the jitter estimate. Also, if an independent estimate of additive noise is available, its effective contribution to amplitude variations can be removed prior to forming the estimate of jitter.

Analyses and graphs as those above represent the condensation of large amounts of data to a few statistics, and important details of digitizer performance can easily be hidden by such a process. Therefore, in developing measurements such as those above, it is suggested that many of the data sequences be plotted along with their Fourier transforms.

## DIGITIZER CALIBRATION AND DISTORTION REMOVAL

Static performance testing may indicate that digitizer calibration is in order. The testing may have turned up a gain or an offset error;

perhaps a slight timing compression. These can most often be adjusted manually, but sometimes it's simpler to automate the procedure using minicomputer control. Even more troublesome, system nonlinearities may have been found during static testing. Correction here is necessarily more complex than that for linearity errors, and digital correction techniques are of proven value.

## CRT Distortion in the R7912 Transient Digitizer

In the case of the R7912 Transient Digitizer, computer correction of static nonlinearities is possible for the entire digitizing instrument or the mainframe only -- the instrument minus plug-in amplifier and timebase. The computer correction applied is essentially the same in either case, but the latter requires no external reference signals and provides corrected system accuracies that are sufficient for many applications.

The R7912 is a digitizer of the scan converter type, consisting of a plug-in amplifier and timebase, a double ended CRT, encoding logic and memory (Fig. 6).

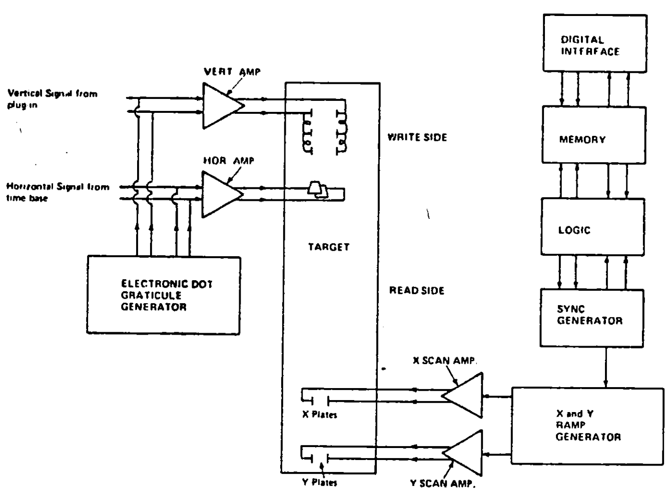



Fig. 6. Block diagram of the R7912 Transient Digitizer. Note the electronic dot graticule generator which provides the reference signal for correcting CRT distortion.

The reference signal used for detection of mainframe distortion is the electronically generated dot graticule (Fig. 7). It is injected directly into the mainframe amplifier, and thus undergoes mainframe distortions (static) identical to that of external signals. Comparing the digitized dot graticule to a computer generated reference graticule defines the two dimensional distortion pattern. Figure 8 provides an exaggerated illustration.

The CRT distortion in the R7912 is fairly typical for modern, high speed tubes. When used

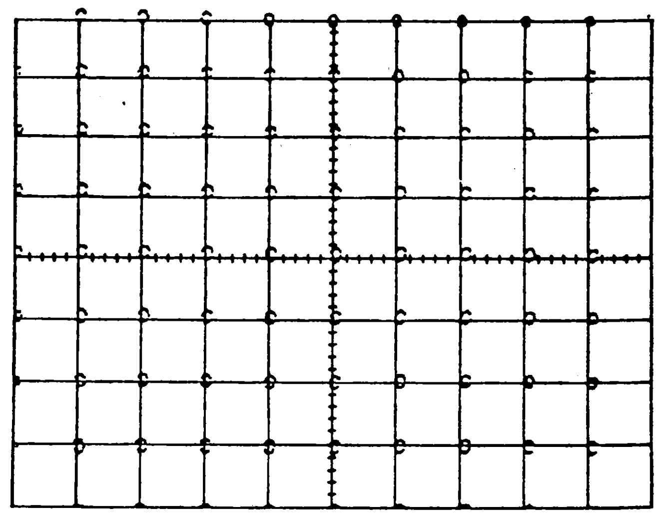


Fig. 7. R7912 dot graticule with computer generated grid overlayed.

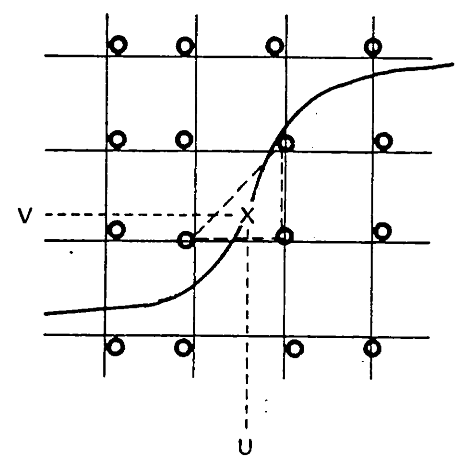



Fig. 8. The electronic dot graticule and the CRT distortions it undergoes are apparent when an exact computer generated graticule is overlayed. This illustration also depicts a point (u,v) on a waveform and the surrounding three graticule points used in generating corrected coordinates.

as an acquisition and viewing instrument, the resulting performance is adequate. But in the digital domain, CRT distortion in the R7912 can generate easily detectable errors. For example, digitizing 10 cycles of a sine wave with peak-to-peak amplitude set at three-fourths of full deflection, can lead to harmonic distortion components in the -30 dB range measured with respect to the fundamental frequency. Thus, correcting for distortion is essential to achieving optimal instrument performance.

## Correcting Distortion

The correction procedure consists of first, acquiring and digitizing the dot graticule to define the distortions. Second, as waveform data is acquired, it is corrected via a two dimensional polynomial interpolation.[10,11] This second step consists of three parts:

1. Locate each waveform point (u,v) within the appropriate triangle, where triangle vertices are the distorted dot graticule points (see Fig. 8). Both u and v are integers, u being a memory address and v a quantized amplitude.
2. Use the distortion errors (dot graticule coordinates minus computer graticule coordinates) at the three vertices to do a three point interpolation, generating corrected points ($u_c$, $v_c$). Store $u_c$ and $v_c$ values in separate arrays.
3. Any distortion in the horizontal dimension implies that the corrected horizontals, $u_c$'s, are both nonintegral and unevenly spaced. The final phase thus consists of using the $u_c$ and $v_c$ values to interpolate to evenly spaced, corrected data.

This correction procedure has been implemented in PDP-11 assembly code and integrated into our BASIC-like signal processing languages.[12] With four BASIC level statements the electronic graticule can be generated, digitized, acquired, and correction tables installed. Thereafter, waveform data can be routinely corrected with four BASIC commands per waveform.

## Results

Comparing uncorrected and corrected waveform data vividly illustrates the effects of "geometry correction." Figures 9 and 10 present the same dc voltage waveform before and after correction. The data of these figures is digital output -- it has not been scaled to volts. Also, the mean level has been removed to better display misalignment and distortion errors. Note that the uncorrected dc level spans three full addresses, while the corrected data spans less than one and one-half addresses.

Amplitude or vertical information in the R7912 is quantized to nine bits (512 levels) and full scale actually corresponds to 480 levels. The quoted vertical resolution of the instrument is 320 lines, and reducing distortions to levels consistent with this quoted resolution leads to optimal performance. The corrected dc level of Fig. 10 with its one and one-half address span roughly corresponds to 480/(3/2) = 320 levels.

Since dc levels do not reflect horizontal distortions, time varying waveforms must be examined too. The obvious choice is a sine wave. The amplitude spectra of Fig. 11 and 12 are of uncorrected and corrected sine waves of 10 cycles per screen and peak-to-peak swing set at three-fourths of full scale. The vertical scale is in dB's relative to the amplitude of the sine wave frequency. Note also the broadening of the harmonic components. This results from a cosine data windowing operation done before Fourier transforming to control leakage.[13] Since this sine wave example, and the dc level test above, involve both an external signal and a plug-in amplifier and timebase, these components were independently examined beforehand to insure the validity of the tests.

In both the dc level and sine wave examples, geometry correction based on the electronic dot graticule significantly reduces systematic errors in the R7912 mainframe.

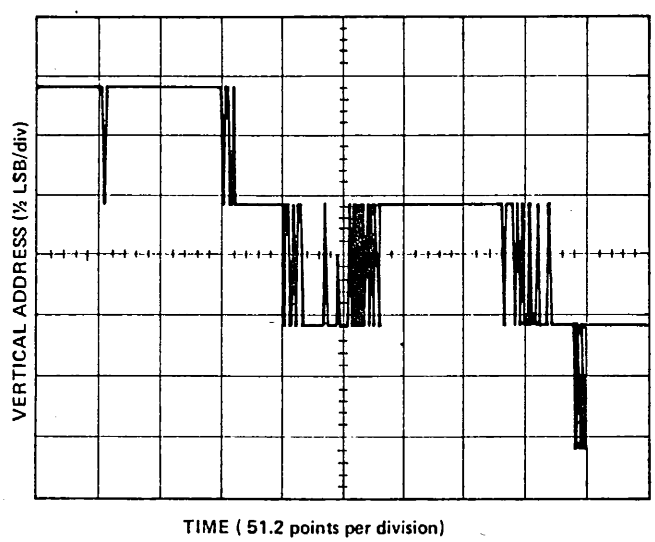



Fig. 9. Precision dc voltage spans three full vertical addresses due to CRT misalignment and distortion. The mean voltage level was removed to more clearly display these variations. (LSB = Least Significant Bit.)

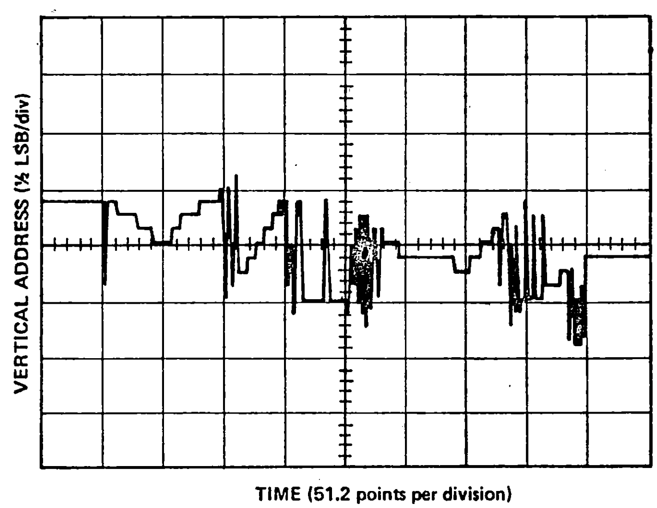



Fig. 10. After applying the correction algorithm, dc voltage of Fig. 8 spans only one and one-half vertical addresses. (LSB = Least Significant Bit.)

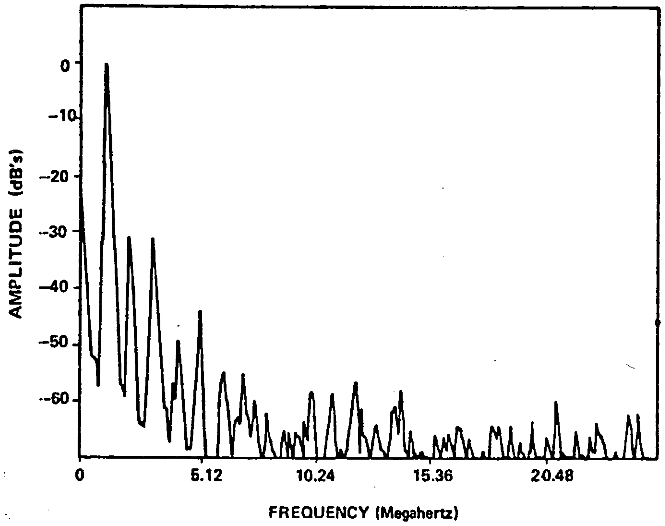



Fig. 11. Amplitude spectrum of 1 MHz sine wave acquired with R7912 Transient Digitizer. Sine wave amplitude was three-fourths of full scale and the sampling rate was 51.2 MHz.

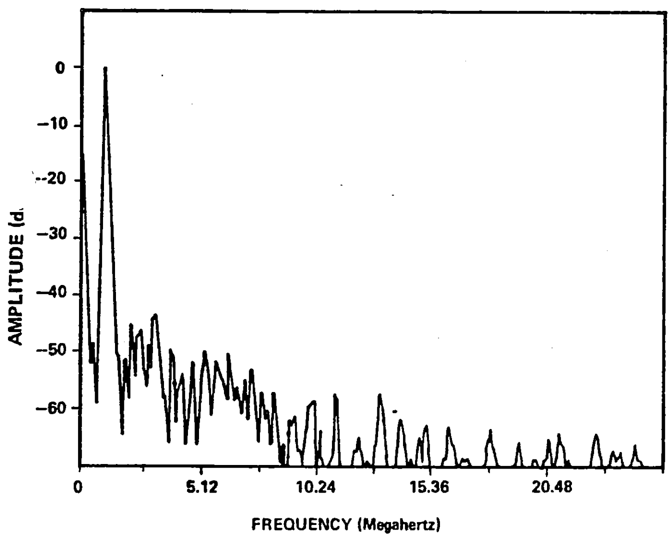



Fig. 12. After applying the correction for CRT distortion to the sine wave data, the distortion components seen in Fig. 10 have been substantially reduced.

## REFERENCES


1. "What's Wrong with A/D Converter Specs," Bernard Gordon, EEE, February 1969.

2. "Don't Lean on A/D Specs," William J. Pratt, Electronic Design, April 12, 1974.

3. "Principles of Data Acquisition and Conversion," Datel Engineering Product Handbook on A/D and D/A Converters, 1974, pp. 6-16.

4. "Critical Design Parameters and Test Methods for Ultra High Speed Analog-to-Digital Conversion, James J. Connolly, et al, Journees D' Electronic, 1973. Conference on Analog to Digital Conversion. Authors with Inter-Computer Electronics, Inc., U.S.A.

5. "Test A/D Converters Digitally," Bill Pratt, Electronic Design, December 6, 1975.

6. "Analog-to-Digital Converter Evaluation by Digital Computer Analysis of Recorded Output," Vol. I, Robert J. Porge, et al, University of Alabama, NTIS Doc. No. AD-775442.

7. Fitting Equations to Data, C. Daniel and F. S. Wood, Wiley-Interscience, 1971.

8. System/360 Scientific Subroutine Package, Version III, H20-0205-3, IBM Corporation.

9. Nonlinear Least Squares Routine, No. 4051/Regression-001, Plot 50, Application Program Library, Tektronix, Inc.

10. Numerical Methods for Scientists and Engineers, R. W. Hamming, McGraw-Hill, 1973, Chapts. 14-15.

11. "Computer Aided Estimation and Correction of CRT Distortion," Lyle Ochs, Tektronix, Inc.

12. WDI Tek BASIC Instruction Manual, PN 070-1774-00, Tektronix, Inc.

13. The FFT: Fundamentals and Concepts, R. W. Ramirez, PN 070-1754-00, Tektronix, Inc.

Vol. 5 No. 1 SPRING/SUMMER 1980

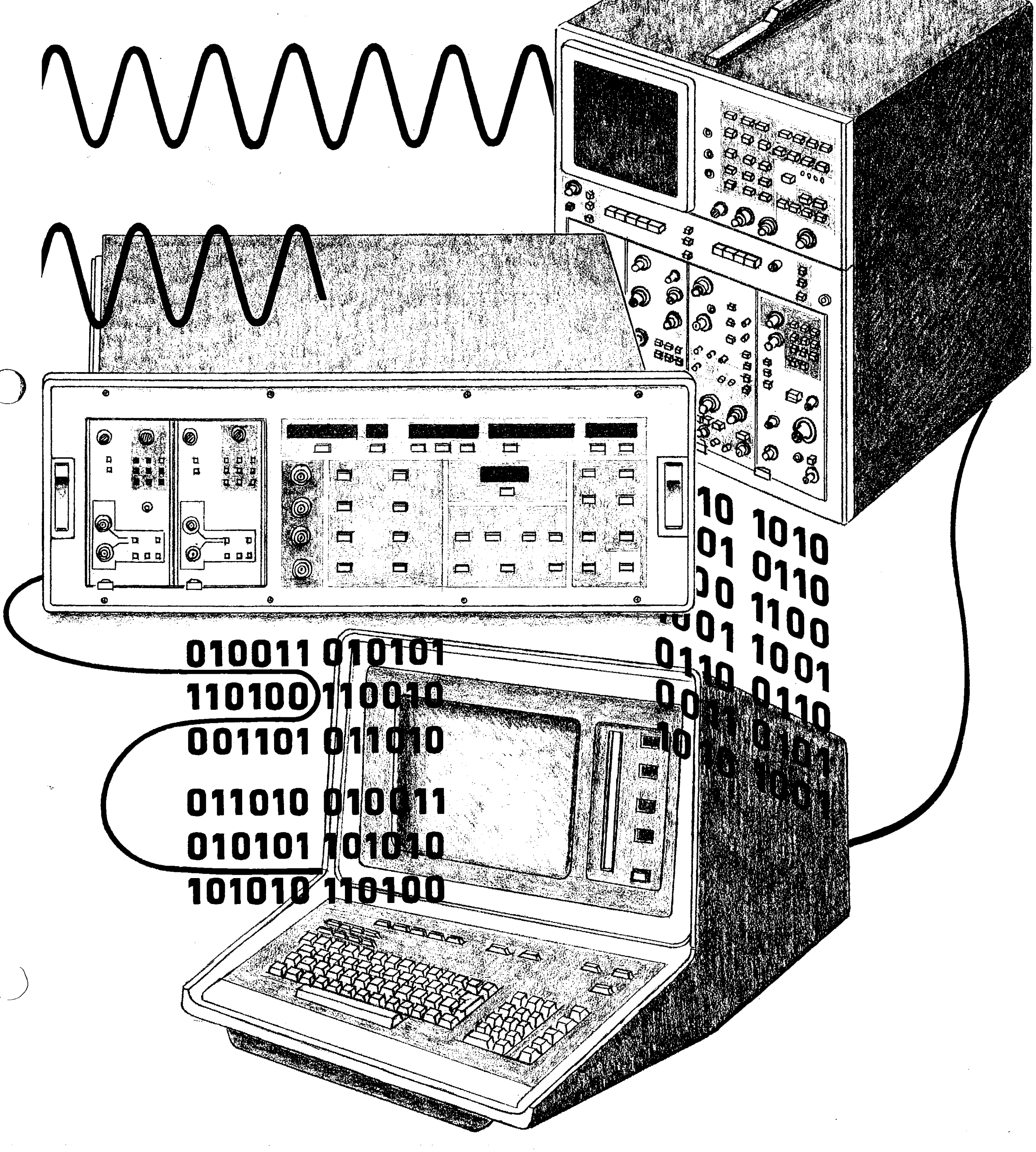
-PAGE 9-

Tektronix®
COMMITTED TO EXCELLENCE

# HANDSHAKE

Newsletter of the Programmable Instruments Users Group

# New Digitizers for a New Decade



Tektronix® COMMITTED TO EXCELLENCE | HANDSHAKE
Newsletter of the Programmable Instruments Users Group

## Table of contents



**Managing Editor:** Bob Ramirez
**Edited by:** Bob Ramirez
**Graphics by:** Bernard Chalumeau

**HANDSHAKE** is published quarterly by the SPS Documentation Group. Permission to reprint material in this publication may be obtained by writing to:

**HANDSHAKE** Editor
Group 157 (94-384)
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077



Printed in U.S.A.

# From R&D to QC...

With a multiplicity of new and exciting features—such as two independent channels, pretrigger, multiple record lengths, and sample rate switching within records—it's difficult to pick a beginning point for describing the 7612D. Perhaps as good a starting point as any is its full name.

Officially this new high-speed (200 MHz sequential sampling rate, 90 MHz bandwidth) instrument is called the 7612D Programmable Digitizer. And, as its name implies, it is fully programmable. Every instrument setting can be controlled by software over a General Purpose Interface Bus (GPIB). Of course, programmability and the GPIB aren't all that new. But they continue to be important to anyone facing the drudgery of repetitive measurements.

What is new, and more important in taming difficult waveforms, is that the 7612D Programmable Digitizer is a dual-channel waveform digitizer—a true dual-channel digitizer with a newly designed analog-to-digital converter for each channel and a crystal-controlled time base for each channel (see Fig. 1). Plus, each channel can be operated independently or dependently at whatever individual time base setting you like. Plus, because of sequential sampling and the 2048 words of partitionable memory provided for each channel, a variety of choices of how to digitize and of how to store each waveform become possible. As examples, you can:

- Capture simultaneous events with the two channels.
- Heighten definition of rapid signal transitions by adjusting sample interval during acquisition.
- Quickly capture and store up to eight successive events per channel by dividing memory.
- Avoid most common triggering problems by using pre- or post-triggering to shift waveform capture time to either side of the trigger point.

Since these features are designed to make short work of what have traditionally been the most difficult or sometimes impossible measurement situations, they bear some further examination.

### Two channels, dozens of possibilities

Operated independently, the dual-channel feature of the 7612D Programmable Digitizer makes the instrument appear as two independent

# New digitizer tames difficult waveforms

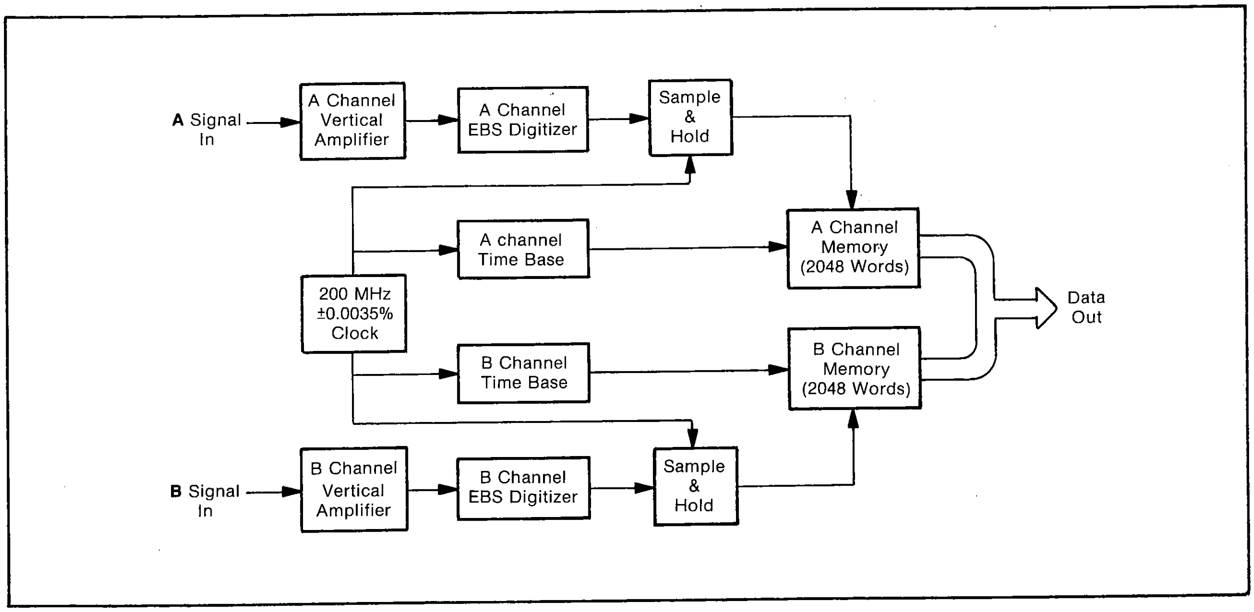



**Fig. 1.** *Diagram of 7612D dual-channel organization. Notice that the sample-and-hold operation follows the EBS (electron-bombarded semiconductor) digitizer. This nontraditional arrangement is part of the state-of-the-art approach of EBS digitizing where analog signals are fed to the digitizer and high-speed sampling is done later with digital comparators.*

waveform acquisition units. This is convenient when you need to deal with two different waveforms or measurements, but it is an absolute necessity if the two waveforms are one-of-a-kind and must be looked at simultaneously. Examples of the latter are capturing input-output information such as in transient response studies, other transient cause-and-effect relationships,

## The EBS secret: How to digitizer faster, more accurately

Waveform digitizing today can be a bed of roses—a fragrant cushion of soft petals or a thorny pile of stems, depending upon what you want to do.

For example, putting digital power and accuracy behind standard oscilloscope measurements has been possible for quite some time. The TEKTRONIX Digital Processing Oscilloscope (DPO) can take you up to 150 MHz, and now the new 7854 Oscilloscope gives you a 400 MHz bandwidth for digitizing waveforms. You can even push these instruments into the GHz region with sampling plug-ins. A bed of roses...as long as you are dealing with repetitive waveforms.

But things can turn thorny for transient signals. You can still comfortably digitize low-speed single-shot phenomena with the psuedorandom techniques of the DPO or 7854. Even very fast transients, requiring up to one gigahertz bandwidth, can be captured and digitized with the scan-conversion technique used in the 7912AD Programmable Digitizer. It's the medium-speed transients that can be a thorn in the side. And they are a particularly sharp thorn when long records (1024 or more points) are necessary for time resolutions on decays or when pre- or post-trigger capabilities are needed. These capabilities can only be provided by high-speed sequential digitizing.

Semiconductor technology has been making strides toward higher frequency sequential digitizers on chips. Current offerings, however, have only reached the 8-bit 30 MHz mark, although faster chips are in experimental stages. Experimental components aside, the only way to push above 30 MHz for sequential real-time digitizing is with a flash converter.

Figure 1 (page 5) shows a typical flash converter layout. The concept is quite simple. Each

*continued on page 5*

and certain voltage breakdown phenomena such as illustrated in Fig. 2.

The two channels of the 7612D can also be operated dependently. For example, one can be triggered after the other to provide a higher sensitivity look at the beginning or ending of a waveform. Or, in another case of one triggered after the other, the full 4096 words of 7612D memory can be used to capture long-duration waveforms such as frequently occur in studies of transient shock and decay phenomena. Figure 3 illustrates both of these uses.

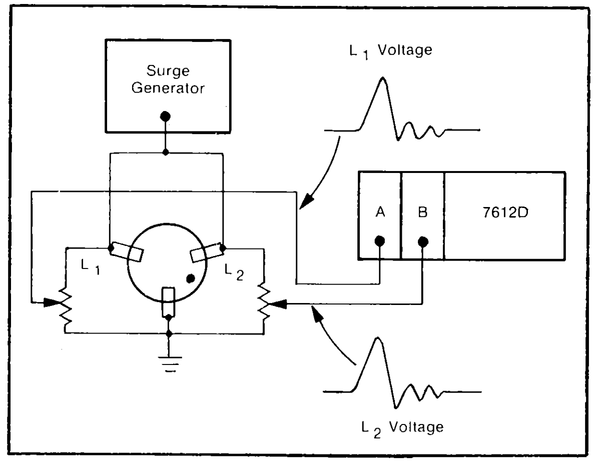



**Fig. 2.** *Dual-channel operation allows capture of simultaneous occurrences. In the above case, a 3-electrode gas lightning arrester with a common discharge chamber is being evaluated for simultaneous and equal breakdown.*

## Multiple records conquer multiple waveforms

While each channel has available up to 2048 words of memory, it may not always be necessary or even desirable to use that full record length for storing a waveform. Consequently, the 7612D Programmable Digitizer has been designed to allow sectioning of each channel's memory into multiple records. You can have one 2048-word record, or up to two 1024-word records, or up to four 512-word records, or up to eight 256-word records for each channel.

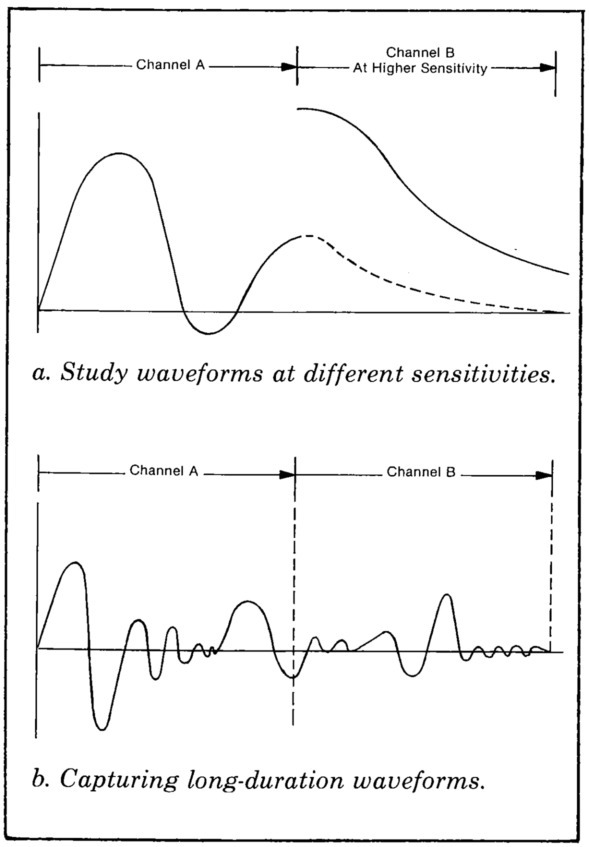



**Fig. 3.** *By triggering the B channel after the A channel, transient decays can be studied at greater sensitivity (a) or long-duration signals can be concatenated into the full 4096 words of available memory.*

When either channel is operated in a multiple-record mode, a trigger is required for digitizing into each record. However as each record is filled, the 7612D trigger circuitry is automatically rearmed. This way acquisition into the next record can begin as soon as a trigger is recognized. The result is that you can acquire a sequence of up to eight 256-point waveforms into each channel, or, if both channels are partitioned to their maximum, that means a total acquisition capability of sixteen 256-point waveforms.

This multiple-record capability with automatic trigger rearming allows you to capture rapid successive events such as indicated in Fig. 4. And, as an additional measure of flexibility, the sample-rate-switching and pre-trigger capabilities can also be used for each record.

## Capture quick changes with sample rate switching

Not only can the sample rate be changed for each channel of data, but it can also be varied within records. This latter case is referred to as sample rate switching, and its effect is similar to

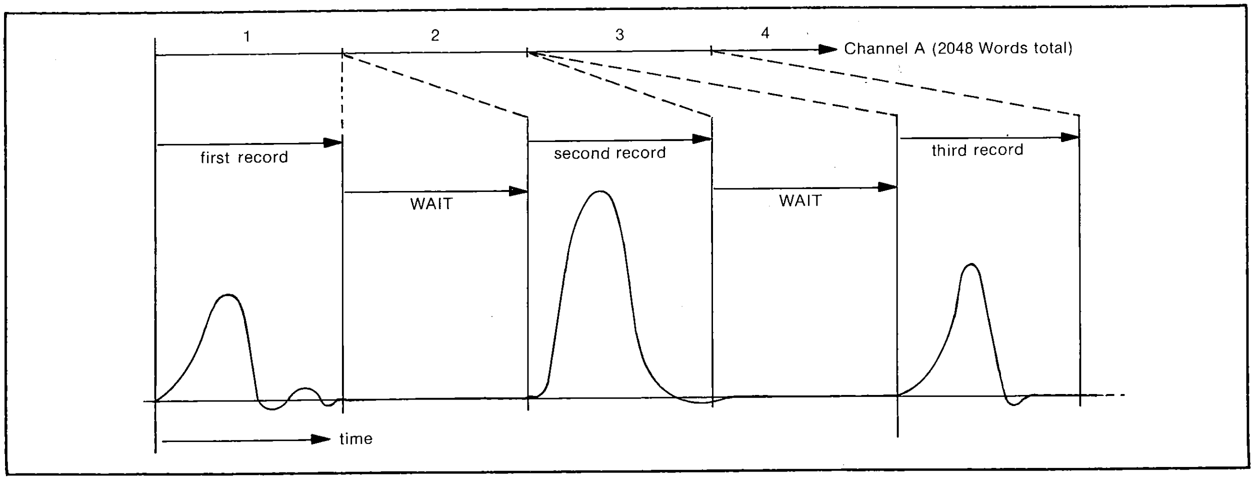



**Fig. 4.** *The multiple-record capability allows capture of a rapid succession of events. After capture, the up to eight waveforms allowed per channel can be transferred to an external device for either storage or complete parameter analysis.*

providing a stepped sweep on an oscilloscope or being able to switch sweep speeds during a sweep.

The capability of sample rate switching is a real boon when dealing with waveforms having fast transitions mixed with slower changing or stable segments. An example is shown in Fig. 5, where fast sampling allows fine definition of the pulse's rise and fall and slower sampling on the pulse top avoids amassing tremendous quantities of redundant data. The available sample intervals derived from the internal clock are selectable in 74 steps (1,2,3,...,9 sequence) from five nanoseconds

*continued on page 7*

## The EBS secret...

comparator in the converter is referenced at an incrementally higher voltage through the precision resistor divider. On the other side, the converters are parallel fed by the signal to be digitized. The converters referenced above the signal level at any point in time remain off. Those referenced below the signal level are turned on, and the highest ON increment is converted to a binary output. It's a very fast technique of conversion, but for n bits of resolution it requires $2^n$-1 comparators plus a very precise voltage divider. That is 255 comparators for an 8-bit flash converter—an expensive approach in terms of number of components, real estate, and number of calibration steps to trim each resistor in the divider.

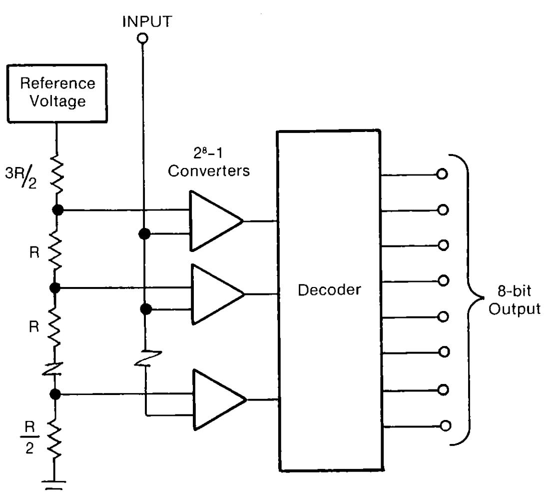



**Fig. 1.** *A typical flash converter circuit requires $2^n$-1 comparators, where n is the desired bits of resolution. For 8 bits, that is 255 comparators.*

The EBS (Electron Bombarded Semiconductor) approach, by contrast, reduces the comparators to a number equal to the bits of digitizer resolution. For an 8-bit digitizer, there are 8 latching comparators as opposed to the 255 amplitude comparators required for traditional flash conversion. The EBS method still enjoys parallel input conversion speed but also uses the cathode ray tube techniques of the scan converter to achieve simplicity and accuracy. The EBS concept embodied by this combination of technologies is illustrated further in Fig. 2.

The EBS tube consists of various focusing rings and alignment plates that converge electrons from the electron gun into a flat ribbon beam. The signal to be digitized is applied to a set of vertical deflection plates that cause the ribbon beam to move up and down according to the applied signal

*continued on page 6*

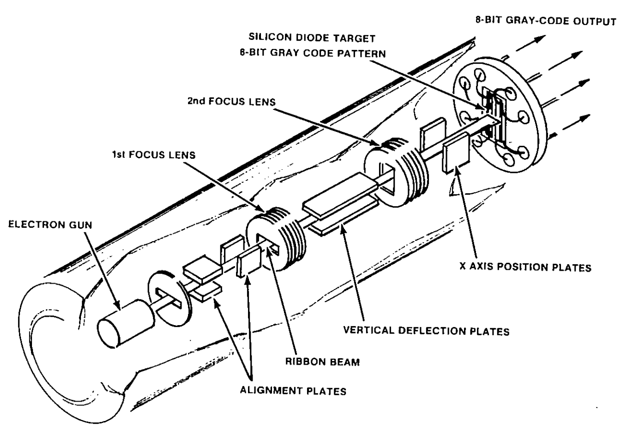
## The EBS secret...

amplitude. A diode target strip, which the beam is focused on, encodes the vertical position of the beam and thus the amplitude of the signal being digitized.

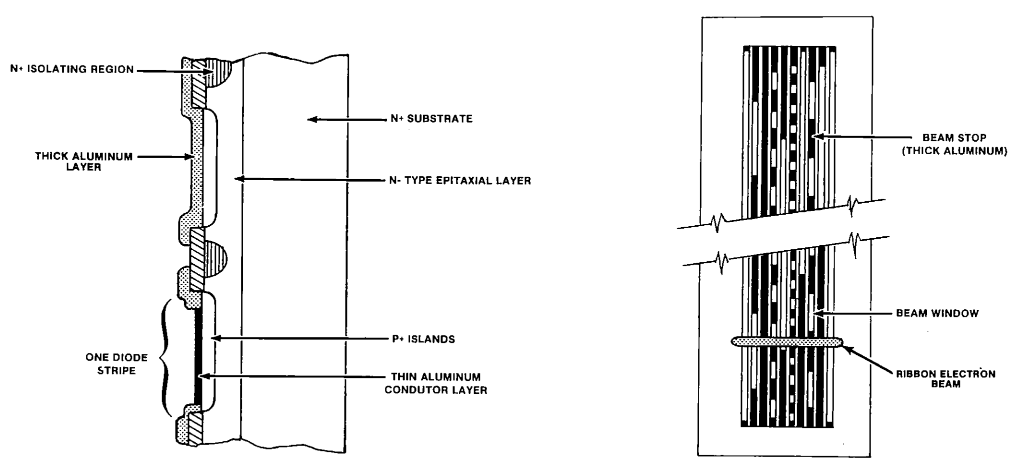
The diode target construction, also shown in Fig. 2, is such that there are 10 diode strips on the target. The two outer diode strips are unmasked over their lengths and serve as beam calibration guides. The inner eight diodes are masked over their lengths leaving only exposed ports or windows positioned to form a gray code indicating vertical position. Whenever the ribbon beam passes through a port, it causes the underlying diode to turn on. The vertical position of the ribbon beam on the target and the arrangement of ports result in the eight diode strips being on or off in a gray-code pattern that uniquely defines the beam position.

The on or off condition of each diode strip is detected by Tektronix designed high-speed comparators, one comparator for each strip. Since the least significant bit coming from the target can be changing at a rate in excess of 2 GHz, signal lead lengths become critical. To keep the leads short, the eight comparators are fastened directly to the outside of the EBS tube near the target. The digitized signal amplitude is clocked out of the tube-mounted comparators at a 5-nanosecond (200 MHz) rate.

Since electron beams can be accurately and rapidly deflected over short distances and diodes can be designed for very fast responses, the EBS technique realizes unprecedented speed in sequential, real-time digitizing. As for accuracy, it's inherent in the precision masking of semiconductor technology, the advanced state of cathode ray tube technology, and the requirement for only 8 comparator latches instead of 255 amplitude-calibrated comparators. That's the EBS secret—combining state-of-the-art technologies to solve the thorniest digitizing problems.




*a. EBS tube construction.*




*b. Diode target details (target size: approximately 250 x 40 mils).*

**Fig. 2.** *The EBS tube used in the 200-MHz, 8-bit 7612D Programmable Digitizer.*

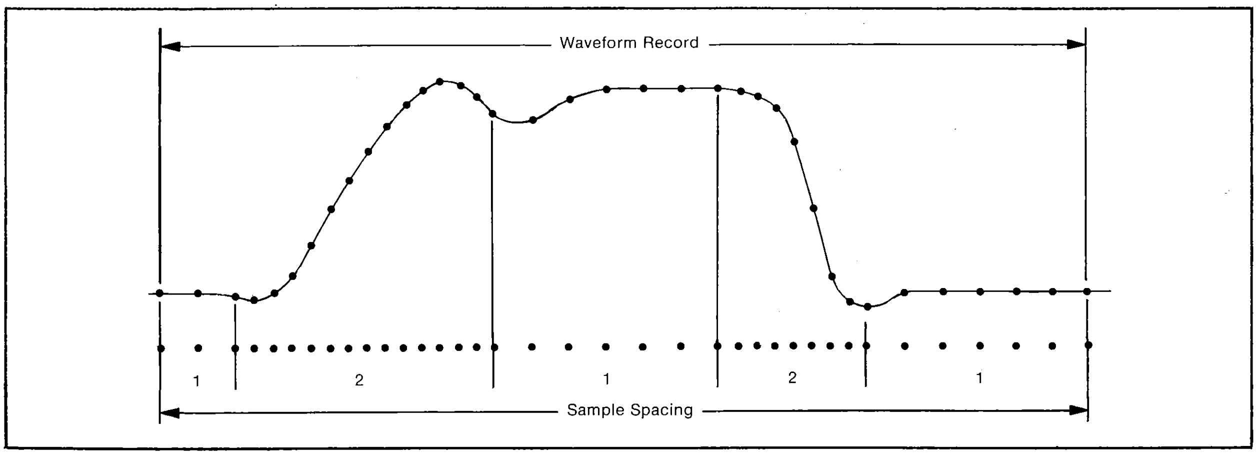



**Fig. 5.** *Sample rate switching during the waveform record can be used to increase time resolution on transitions. The above example shows switching between rates 1 and 2 with four rate changes during the record. Up to 13 sample rate changes are allowed during a waveform record.*

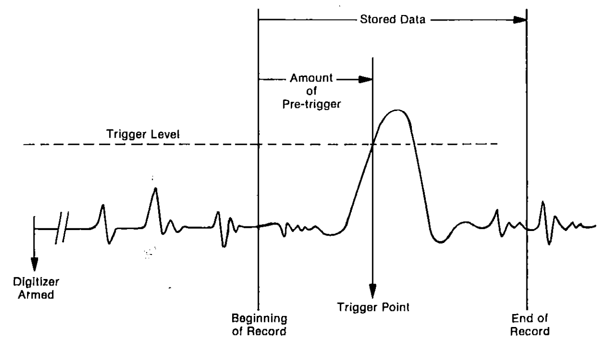
to one second. Switching intervals or sample rates can be done up to 13 times during a waveform record.

## Pre- and post-trigger improve data choices

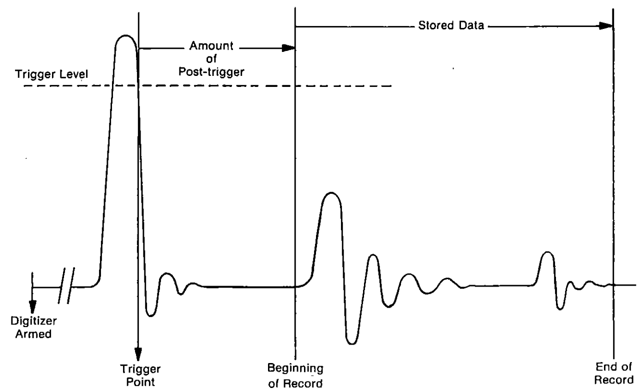
The pre-trigger and post-trigger features of the 7612D Programmable Digitizer allow you to effectively advance or delay waveform acquisition from the point of trigger. Either operation stems from the continuous, sequential sampling provided by the 7612D. Once the digitizer is armed, it begins digitizing samples sequentially and storing them in its first-in, first-out memory. New waveform samples are pushed into one end of the memory while the oldest samples are discarded as they are pushed out of the other end. This goes on until a trigger is received and implemented. Implementing the trigger causes the current set of waveform samples to be frozen in memory. The operations of pre-trigger and post-trigger are obtained simply by digitally advancing or delaying trigger implementation. The affects of this are shown in Fig. 6.

In Fig. 6a, the use of pre-trigger to get a stable trigger above base-level noise is illustrated. The trigger level is set high enough to obtain stable triggering on the waveform. At the same time, the pre-trigger time is set to include storing of a selected portion of data occurring before the trigger point. The result is that you can avoid noisy triggering and still get waveform data that includes the entire leading edge as well as data preceding the leading edge if you like.

Another use of pre-trigger is in the study of cause and effect. The effect might be the easiest or only thing to trigger on, in which case, pre-trigger lets you look back in time at the cause. Studying transient build ups associated with system power-on sequences is just one example of this.




*a. Pre-trigger operation.*




*b. Post-trigger operation.*

**Fig. 6.** *Pre- and post-trigger operation let you select just the segment of data you want.*

The opposite situation, post-trigger operation, is shown in Fig. 6b. In this case, fixing the stored data in memory doesn't occur until the selected post-trigger time has elapsed. This is useful for

# From R&D TO QC...

acquiring data occurring some time after a triggering event—aftershocks or reverberations resulting from a test impulse, for one example.

Then of course there is standard triggering, too, where storage starts at the time of triggering. So you can move any way you like through time to get just the right data for your application.

### And it's all programmable

The amount of pre- or post-trigger, sample interval and where it should be switched and by how much, record length and number of records, and triggering modes for independent or dependent dual-channel operation can all be set or selected either manually or under program control over the General Purpose Interface Bus (IEEE 488-1975). And, beyond these major features, all of the other mainframe settings (except power ON-OFF) can be controlled either manually or under program control. Add to this a pair of 7A16P Programmable Amplifiers for the vertical channels and the instrument becomes a totally programmable instrument.

Not only is the 7612D programmable in the sense that software can be used to change instrument settings, but it is also programmable in the sense that software can read or learn current instrument settings. You can work out a test sequence manually the first time while software learns and stores the settings. Then the next time the test is run, it can be run completely under software control, including setting up the instrument.

General instrument control routines can be written for exploring the unknown or uncertain in R&D. Or more specific routines can be written for quickly executing the repetitive operations common to QC applications. In either case, the 7612D Programmable Digitizer has the power and flexibility to freeze even the most unruly waveforms in memory. Add to this the signal processing power offered by TEK SPS BASIC software, and those waveforms can be analyzed in detail for whatever parameters your application demands.

*By Bob Ramirez*
*HANDSHAKE Staff*

# Interactive 7612D acquisition routine

This bare-bones program was written by Mark Tilden, Tektronix SPS Documentation Group, to interactively control one 7612D via the TEK SPS BASIC low-level GPIB driver (GPI.SPS). In response to the program's prompt (**7612**>), you can ask for the content of the status byte or enter a single 7612D set or query command. The program also responds to an SRQ interrupt from the 7612D by reporting the content of the status byte. No checking is made to see if the input is correct or appropiate, so careless typing may cause an error. If an error is fatal, you should restart the program after sending the 7612D a device clear.

```
10 LOAD "GPI
20 SIFTO @0,6000
30 PRINT "7612>";
40 WHEN @0 HAS "SRQ" GOSUB 240
50 INPREQ GOSUB 70
60 GOTO 60
70 INPUT C$
80 IF C$="STAT?" THEN 210
90 PUT C$ INTO @0,32,96
100 L=LEN(C$)
110 IF SEG(C$,L-5,L-2)="READ" THEN 160
120 IF SEG(C$,L,L)<>"?" THEN 270
130 GET A$ FROM @0,64,96
140 PRINT A$
150 GOTO 270
160 DELETE A
170 READBI A FROM @0,64,96
180 PAGE
190 GRAPH A
200 GOTO 270
210 GETSTA @0,ST,64,96
220 HPRINT "MAINFRAME STATUS (HEX): ";ST
230 GOTO 270
240 POLL @0,ST,PA,SA;64,96;64,97;64,98
250 PRINT "SRQ FROM ADDRESS :";PA;SA;
260 HPRINT " STATUS (HEX):";ST
270 PRINT "7612>";
280 RETURN
```

For expediency, the routine assumes the interface number, the primary address, and the secondary address are all zero. This means a primary listen address of 32, a primary talk address of 64, and a secondary address of 96 are used with the low-level driver.

The first six lines set up the program. The driver is loaded and the time-out value is set to six seconds, and then the prompt is printed. Next, an SRQ interrupt and an input request are set up. Finally, the program loops at line 60 until a line is entered from the keyboard or an SRQ is asserted. When either happens, control transfers to the subroutine which handles the 7612D command string or prints the status byte and prints the prompt when done. Then control returns to line 60 where the program continues to loop.

All input (except the string "STAT?") is simply

*continued on page 12*

# Dynamic testing reveals overall digitizer performance

Typically, full bit resolution and fastest sampling rate are the first two characteristics given when real-time digitizer performance is mentioned. These two items head the specification list and, in fact, are important indicators for comparing the ideal performance of real-time digitizers. But, for actual performance in a particular application, a careful look has to be taken at some items further down on the specification sheet.

Further down on the list are such items as gain error, offset error, linearity, and monotonicity. Gain and offset errors manifest themselves as an incorrect signal amplitude and an incorrect DC component, respectively. However, instrument calibration can reduce these errors, and often what's left can be characterized and removed by either firmware or software routines. Linearity and monotonicity errors, on the other hand, cannot generally be calibrated out. Thus, they must be specified individually, or their effects must be considered or accounted for in some other specified parameter.

Still further down on the list, bandwidth might be specified. Unlike sampling rate, which tells you by the Nyquist criterion the highest frequency component ($f_N = f_S/2$) that can be digitized without aliasing errors, the bandwidth specification tells you at what frequency you can expect amplitude attenuation to begin increasing.

Of all the items on the list, which are most important in choosing a real-time waveform digitizer?

## Current specifications only partial picture

All of the standard digitizer specifications are important to consider. But still it is difficult to combine their individual effects into something giving an accurate picture of overall waveform digitizer performance.

To further complicate the issue, there are other sources of error that can be digitizer dependent as well as dependent upon signal frequency or rate of change. Examples of these sources are aperture uncertainty and the response of the least-significant-bit comparators and associated circuitry. Often these error sources are not specified. So the user is left, at best, questioning just how accurately the digitizer performs across the frequency band, and at worst, believing that accuracy corresponding to full resolution is achieved no matter what the input frequency.

The question then arises: "How can all of these errors be taken into consideration when trying to assess actual digitizer performance?"

Efforts to answer this question have resulted in development of the "Dynamic Performance Test." This test is a computer-oriented black-box approach to testing the overall performance of most real-time analog-to-digital converters and waveform digitizers. Its main advantage is consolidation into a single specification most of the otherwise difficult-to-describe error sources.

## Focusing on dynamic operation

Dynamic performance means, essentially, performance under a changing circumstance. In the case of a waveform digitizer, this means testing with a changing input signal, a sine wave for example. In particular, using a sine wave offers several advantages over other types of test signals. First, high-frequency sine waves can be generated with relative ease. And, second, sine waves are easily characterized.

These factors combine to make dynamic testing quite simple in concept. A high-quality sine wave is captured by the digitizer under test. In conjunction with this, an associated computer is used to generate an ideal sine wave matching the captured sine wave in amplitude, frequency, phase, and vertical position (including offset). Also, an ideally sampled and digitized version of the sine wave is generated by the computer. Then the actual digitizer output, the ideal sine wave, and the ideally sampled and digitized sine wave are compared in various ways, and some statistics are computed to reflect the accuracy of the digitizer's output.

That's the basic concept. The details are better revealed through a test example.

To begin, a high-quality sine wave generator is required. Its performance must exceed that of the digitizer under test so that inaccuracies in the digitizer's output cannot be attributed to the signal generator. In some cases, notch filters may be required at the generator's output to keep harmonics below what is expected from the digitizer.

# Dynamic testing...

The output of the sine wave generator is digitized by the instrument under test, and the result stored in the computer. Figure 1 shows an example of stored output from an 8-bit digitizer. Each of the 32 samples shown is an integer corresponding to a quantization level within the range of 0 to 255 (an 8-bit digitizer has $2^8 = 256$ possible levels with zero being the lowest level).

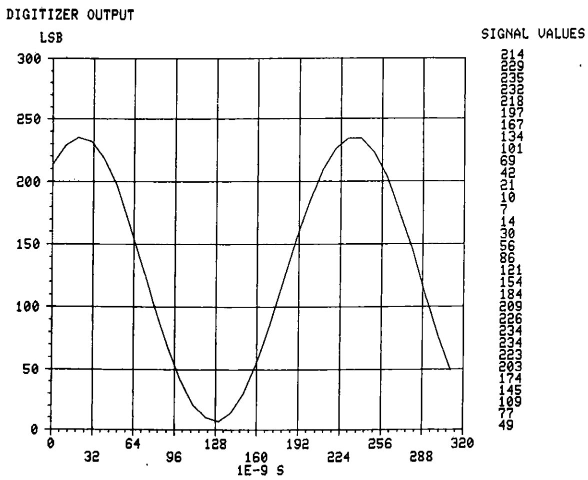



**Fig. 1.** *Stored data from a sine wave that has been sampled 32 times and digitized by an 8-bit digitizer.*

With the test sine wave digitized and stored, computer analysis can begin. The first step is to fit a perfect sine wave to the digitized signal. This perfect sine wave is fully described by

$$A \sin(2\pi ft + \Phi) + C$$

where A is amplitude, f is frequency, $\Phi$ is phase, t is time, and C is DC offset. Fitting this sine wave to the digitized signal is done by a software routine using a least squares method, and the result is considered to be a description of the analog input to the digitizer. It should be noted, however, that because the analog signal parameters are computed from the digitizer's output, DC offset and gain errors are not included. Therefore, tests for DC offset and gain error should be done separately.

Having computed the characteristics of the analog input from the digitizer's output, some statistics necessary to describe the digitizer's performance can be generated. To do this, the computed sine wave input is perfectly sampled by the computer at specified sampling times, t. Values resulting from this are shown in Fig. 2. These values are shown as noninteger values

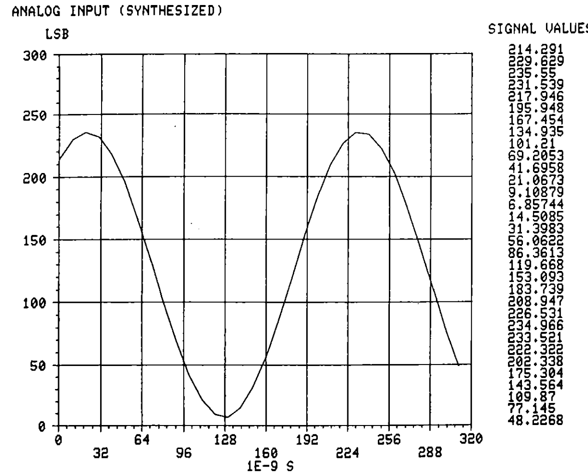



**Fig. 2.** *Ideal sine wave synthesized from output data of the digitizer under test and perfectly sampled at 32 points.*

since only sampling—not quantization—has taken place. Quantization is the next step undertaken by the computer. This is done by rounding the sampled values to the nearest digital level within the proper digitizing range. Figure 3 shows the effects of this perfect quantization. Notice that the values of the computed input signal are now integers within the range of 0 to 255. This (Fig. 3) is ideal digitization of the input signal, or, in other words, the output of a perfect 8-bit digitizer.

The difference between the perfectly sampled signal of Fig. 2 and the real digitizer's output (Fig. 1) represents the analog error signal for the digitizer under test. The difference between the perfectly sampled signal (Fig. 2) and the perfectly

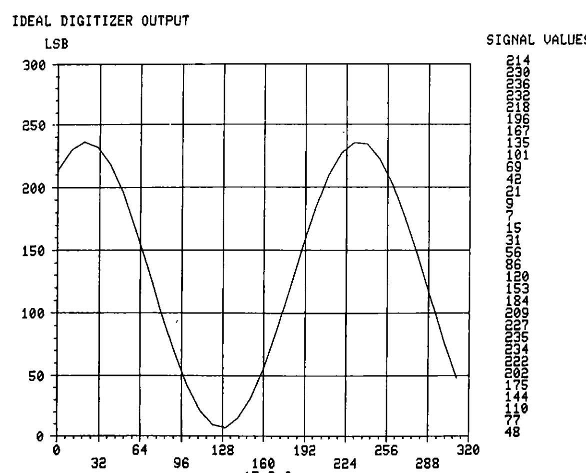



**Fig. 3.** *Synthesized sine wave after being perfectly quantized.*

quantized signal (Fig. 3) represents the analog error signal of the corresponding ideal digitizer. Graphs of these analog error signals for the sample test are shown in Fig. 4. Notice that the values for the errors of the ideal digitizer lie between ±0.5 least significant bits (LSB). This is what is expected for an ideal digitizer. The real digitizer's error values in Fig. 4, however, are greater than ±0.5 LSB. This indicates a loss of accuracy in the digitizing process.

## Developing final results

Two important performance parameters can be computed from the information gathered thus far—the digitizer's signal-to-noise ratio and the number of effective bits of accuracy. The signal-to-noise ratio (SNR) is determined by the following formula:

$$\text{SNR} = 20 \log \frac{\text{RMS (analog signal)}}{\text{RMS (real digitizer errors)}}$$

The effective number of bits lost by the digitizer is given by:

$$\text{Lost Bits (LB)} = \log_2 \frac{\text{RMS (real digitizer errors)}}{\text{RMS (ideal digitizer errors)}}$$

Subtracting the lost bits from the total bits of resolution available yields the number of bits at which the digitizer is effectively operating. In the example used thus far, the digitizer's ideal resolution is 8 bits. So the number of effective bits is then:

$$\text{Effective Bits (EB)} = 8 - \text{LB}$$

Figure 5 illustrates these test results for the example. Besides SNR and effective bits, some

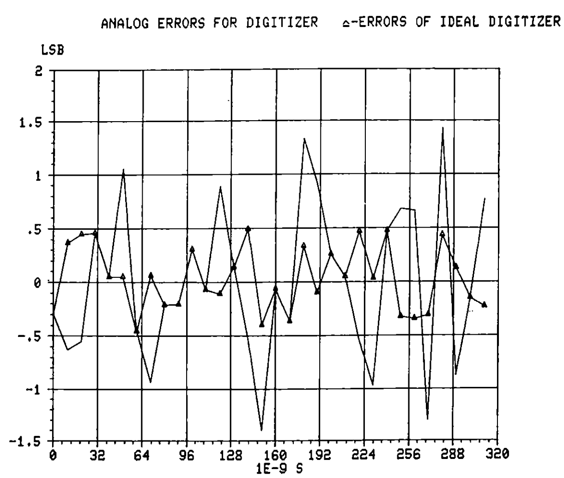



**Fig. 4.** *Example error signals associated with ideal and real 8-bit digitizing.*

```
I.D.  2-14-79 500OHM 4.7MHZ +1.5DB TBC/2
SIGNAL TYPE:  SINE
FILE NAME:  TST3

******** SUMMARY STATISTICS (IN LSB OR AS INDICATED) **

RMS OF DIGITIZER OUTPUT(DC REMOVED) =  79.3016

ANALOG ERRORS FOR DIGITIZER OUTPUT:
      MAX =  1.43629
      MIN = -1.39833
      RMS =  .721989

SIGNAL-TO-NOISE RATIO =  40.8147 DB

RMS ERROR FOR IDEAL DIGITIZER =  .297613

EFFECTIVE BITS =  6.72146
```

**Fig. 5.** *Output of dynamic performance routine.*

other potentially useful performance parameters are also displayed. These are the maximum, minimum, and RMS values for the errors from the digitizer under test and the RMS value of the errors from the synthesized ideal digitizer.

For the example test, Fig. 5 shows the number of bits at which the digitizer was effectively digitizing the 4.7 MHz sine wave to be 6.72146 bits. This means that for the sine wave the 8-bit digitizer was actually operating equivalent to an ideal 6.72146-bit digitizer.

As indicated earlier, this performance degrades as frequency increases. In fact, to get an accurate picture of the digitizer's response, tests at several frequencies should be conducted. The data from this can then be used to construct a graph, such as shown in Fig. 6, for either effective bits or signal-to-noise ratio.

Before testing at several frequencies, however, it should be pointed out that performance is also affected by the amplitude of the test sine wave. Several errors, including aperture error, can arise from the rate of change of the sine wave. This rate of change (dv/dt) depends upon both the frequency and amplitude of the test signal as can be seen from the following relationship:

$$dv/dt = d(A \sin(2\pi ft))/dt = A \cos(2\pi ft)$$

So it is important to keep the sine wave amplitude constant while running a series of tests at various frequencies.

Exactly what amplitude should be used in testing is another subject of discussion. It might be desirable to use an amplitude corresponding to the full resolution of the digitizer. Doing this,

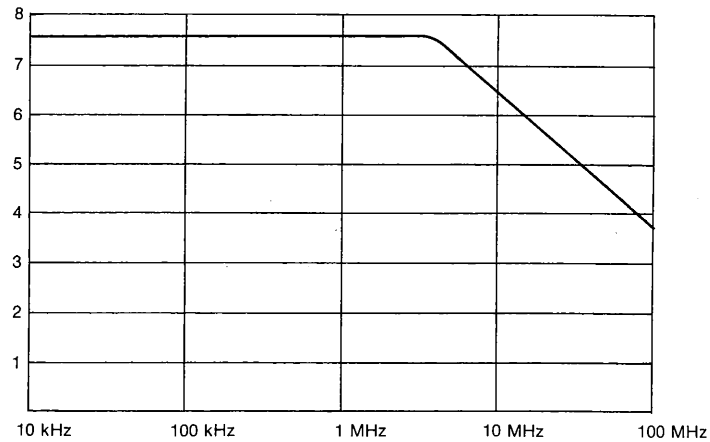



**Fig. 6.** *Plot of dynamic accuracy constructed from sine wave tests at several frequencies.*

however, forces the digitizer to handle the maximum dv/dt for any given frequency. This may not be a really fair test since most waveform digitizers are not typically used in modes where the acquired signal covers the full input scale. In practice, waveforms are more often acquired at less than full scale, so something like a half-scale amplitude sine wave would probably give a better picture of what digitizer performance will be in actual practice.

In fact, the 8-bit TEKTRONIX 7612D Programmable Digitizer has been characterized in this half-scale sine wave manner. The results are listed in the instrument specifications under "Dynamic Accuracy for Digitizing and Storage" and appear as follows:

| Sine Wave Freq. | SNR | Effective Bits |
|---|---|---|
| 300 kHz | 42 dB | 7.8 |
| 20 MHz | 32 dB | 6.0 |
| 80 MHz | 20 dB | 4.0 |

## A better overall picture

With dynamic accuracy specified at enough points, it is possible to construct a graph (such as Fig. 6) and perceive expected digitizer performance in the frequency range of interest. Most error sources (except gain and offset) are accounted for in a few easy-to-understand, easy-to-visualize numbers. There is no longer any need to agonize over combining diverse error figures in an attempt to come up with an overall figure of performance. Indeed, most traditional waveform digitizer specifications become necessary only as a matter of general interest, while dynamic accuracy becomes the primary criterion of performance.

*By Laurie DeWitt, Tektronix, Inc.*
*SPS Signal Analysis Group*

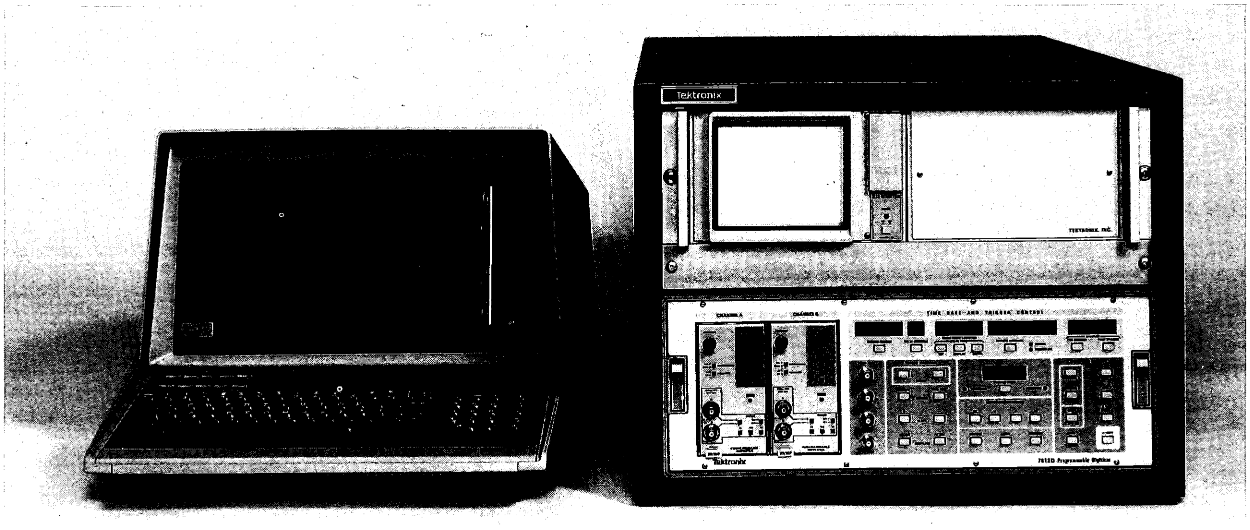
# 7612D acquisition routine...

sent to the 7612D with a PUT statement (line 90). If the entry ends with a question mark, it is assumed to be a query command; so, the response from the 7612D is read into a string variable with a GET statement and printed (lines 30 and 40). If the input does not end in a question mark, it is assumed to be a set command and no further action is taken unless it is a READ. In this case, a single channel of data is read and graphed (lines 160 to 190). No provision is made to handle the ALTernate or REPeat commands or to READ both channels.

If the input is "STAT?", the status byte of the 7612D is read and printed in hexidecimal (lines 210 and 220). Similarly, when an SRQ interrupt occurs, the status byte is read and printed (lines 240 and 250). If you do not have the High-Level Support package, change the HPRINT statements to PRINT statements.

*Joyce Ferriss,*
*HANDSHAKE Staff*

# Configuring the 7612D for measurement solutions



*The WP3110—desk-top economy with minicomputer performance.*

Crystal-accurate, dual-channel digitizing to 100 megahertz. Pre-trigger, post-trigger, sample rate switching, and memory partitioning. The 7612D has them all...all of the capabilities necessary for quick, efficient, and complete waveform capture.

Plus, the 7612D is GPIB compatible and fully programmable. Interface it with a computer, and you have a system powerful enough to smash through any measurement bottleneck.

But wait! Before you pick just any GPIB controller for your 7612D, consider these questions:

- Does the computer have enough memory for two, three... maybe four or more waveform arrays of 2 K-words each?
- Is the data transfer speed adequate?
- Can the system handle arrays...integrate them, differentiate them, fast Fourier transform them?
- What about graphics and hard copy output?
- And what about the experience and expertise necessary for successful system integration?

If you don't have the right answers for these questions, then there's a good chance you'll be missing some important system benefits offered by the 7612D Programmable Digitizer.

A quick, cost-effective way to get the right answers is to look at the WP3000-Series of TEKTRONIX Signal Processing Systems. These fully integrated systems are assembled, tested, and completely documented to give you ready-to-go service from your 7612D Programmable Digitizer.

## Desk-top computer-based system

The economical WP3110 is a pairing of the 7612D with the recently introduced TEKTRONIX 4052 Graphic Computing System. The 4052 is one of the most powerful desk-top computers currently available. Its bit-slice technology challenges minicomputer speeds, and its 64 K-word memory is quite capable of handling the large waveform arrays sent by the 7612D.

Easily programmed in an extended BASIC language, the 4052 offers complete signal acquisition, processing, and display capabilities. Measurement results appear before you in a matter of seconds. Plus, your computational power can be extended over what's available in more conventional desk-top systems by specialized ROM Packs. These ROM Packs both simplify and speed the execution of commonly encountered array processing functions. Things like integration, differentiation, and searches for array maximum and minimum values are carried out by calling single commands. Even the fast Fourier transform is available as a single comand that executes at minicomputer speeds.

Added to these standard processing tools are a variety of already written programs. As a 4052 user, you have access to the PLOT 50 software library, a collection of statistics, mathematics, and graphics programs that are applications oriented. Plus, you automatically become a member of the 4050 Users Club with access to applications software written by other members. Your specific applications program may already be there for the asking!

And once you've run your program, you'll have the advantage of high-resolution 4052 graphics and alphanumeric display. Special graphic routines help you quickly format results to meet your particular needs. Add a 4631 Hard Copy Unit or 4662 Digital Plotter, and your high-resolution displays can be transferred to paper for clearer, more concise scientific reports or quality control documents.

## Minicomputer-based system

The WP3201, a pairing of the 7612D with a DEC PDP-11/34 minicomputer, is at the other end of the waveform processing spectrum. When speed and the amount of information to be handled are primary considerations, the WP3201 offers premium performance.

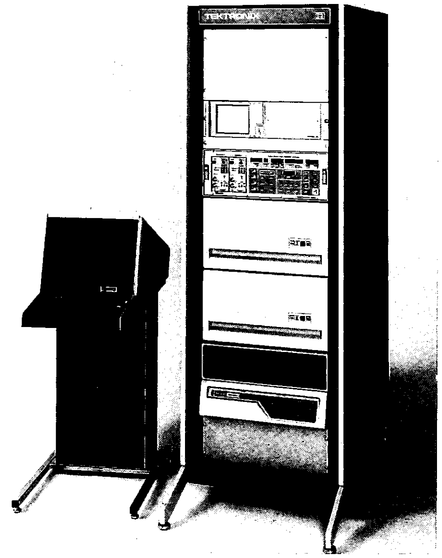
Based on Digital Equipment Corporation's reputable PDP-11/34, the WP3201 offers 128K words of minicomputer memory, two high-capacity disk drives (5 Megabytes each), a graphic terminal, and the field-proven TEK SPS BASIC language with extended memory capabilities for handling the large arrays coming from your 7612D Programmable Digitizer.

TEK SPS BASIC software is like no other software you've seen. It's system software...it's instrument control software...it's signal processing software...it's ready-to-use software.

Designed from an instrument and measurement background, TEK SPS BASIC is aimed at solving measurement problems quickly and completely, with a maximum of ease. The GRAPH command, for example, causes a complete waveform with axes and scale factors to be displayed on the graphic terminal. Complete waveform graphics with a single command!

And there are single commands for many other operations—integration, differentiation, fast Fourier transformation, and many, many more. Plus there are GPIB drivers to ease the burden of instrument control, whether you are using one instrument or a busful.

The fast GPIB driver allows acquisition of 7612D waveforms in times expressed in tens of




*The WP3201—for when speed and memory are of the essence.*

milliseconds. And, with the new DLOG command, one hundred 256-point waveform arrays per second can be logged from a 7612D onto the peripheral disks. These powerful disk peripherals not only allow rapid mass waveform and program storage, but they allow almost instantaneous recall of either data or programs for further processing or display. Display is via a graphic terminal and is implemented through the complete and highly flexible TEK SPS BASIC graphics package. This allows quick and easy formatting of measurement results, and, by adding a 4631 Hard Copy Unit, the display can be transferred to paper for permanent records and reports.

All of this comes assembled and ready to run with on-site installation and system checkout provided as part of the WP3201 package.

## The final question

All of the system questions about adequate memory, data transfer, array processing, graphics, and system integration are answered by the WP3110 and WP3201 systems. This leaves you with the final question: "Which system is best for my needs?"

To help you answer that question, Tektronix maintains a staff of experienced systems specialists in the field. They'll be able to work with you in assessing your measurement needs and choosing the system that is right for meeting those needs. To get in touch with your systems specialist, contact your local Tektronix Field Office. Outside of the United States, contact the Tektronix subsidiary or distributor in your country.

*By Jean-Claude Balland,*
*SPS Marketing Program Manager,*
*Tektronix, Inc.*

# WP1310 backs 400 MHz oscilloscope measurements with computing power

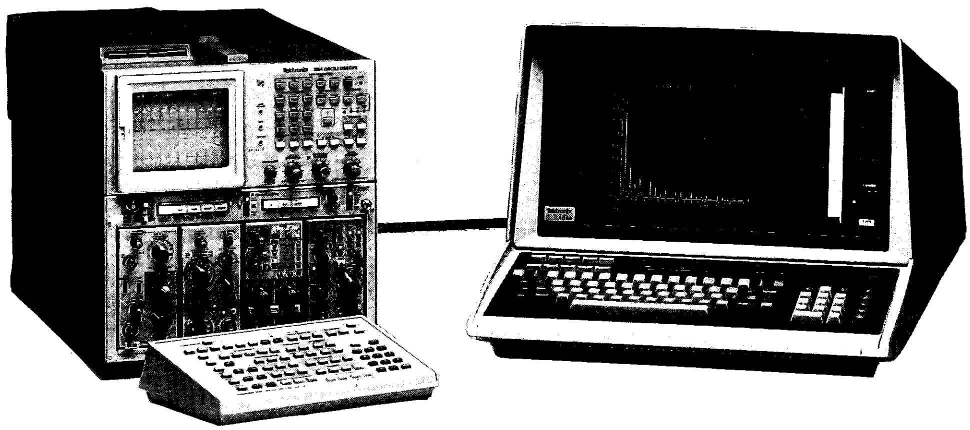


**Fig. 1.** *The TEKTRONIX WP1310 interfaces the 7854 Oscilloscope with the 4052 Graphic Computing System for a combination of modern digital oscillography with BASIC language waveform processing and graphics capabilities.*

Oscilloscope measurements with a new level of flexibility and power—that's the purpose of the WP1310 Waveform Processing System shown in Fig. 1. This new system uses the IEEE 488 bus to tie desk-top computing power to the recently developed 7854 Oscilloscope. The result is a new measurement tool offering you anything from standard oscilloscope measurements, through push-button measurement of pulse parameters, to programmed waveform analyses including operations such as fast Fourier transformations, convolution, and correlation.

## Putting power into oscilloscope measurements

The system drawing in Fig. 2 points out some of the many WP1310 features. It also identifies the major system components, an acquisition unit (7854 Oscilloscope) and a system controller (4052 Graphic Computing System).

Basically, system operation begins with the acquisition unit, the 7854 Oscilloscope. This 400 MHz oscilloscope is not like your usual oscilloscope. Although it can be operated just like an oscilloscope, the WP1310 acquisition unit also contains a waveform digitizer, memory for storing digitized waveforms, and microprocessor power for some standard waveform calculations. As part of the WP1310 system, its function is to capture and digitize your waveforms, speedily preprocess them when necessary, and hand them to the WP1310 system controller for further analysis.

The WP1310 system controller provides high-speed processing as well as instrument control under BASIC language programs. Additionally, with the Signal Processing ROM Packs installed, the WP1310 system controller extends your selection of waveform analysis tools to include such things as

- Data windowing
- Fast Fourier transformation (FFT)
- Inverse Fourier transformation (IFT)
- Auto- and cross-correlation
- Convolution

Plus, the high-resolution graphics capability of the WP1310 allows formatting of results to your specific application needs. Would you like

- Bode plots

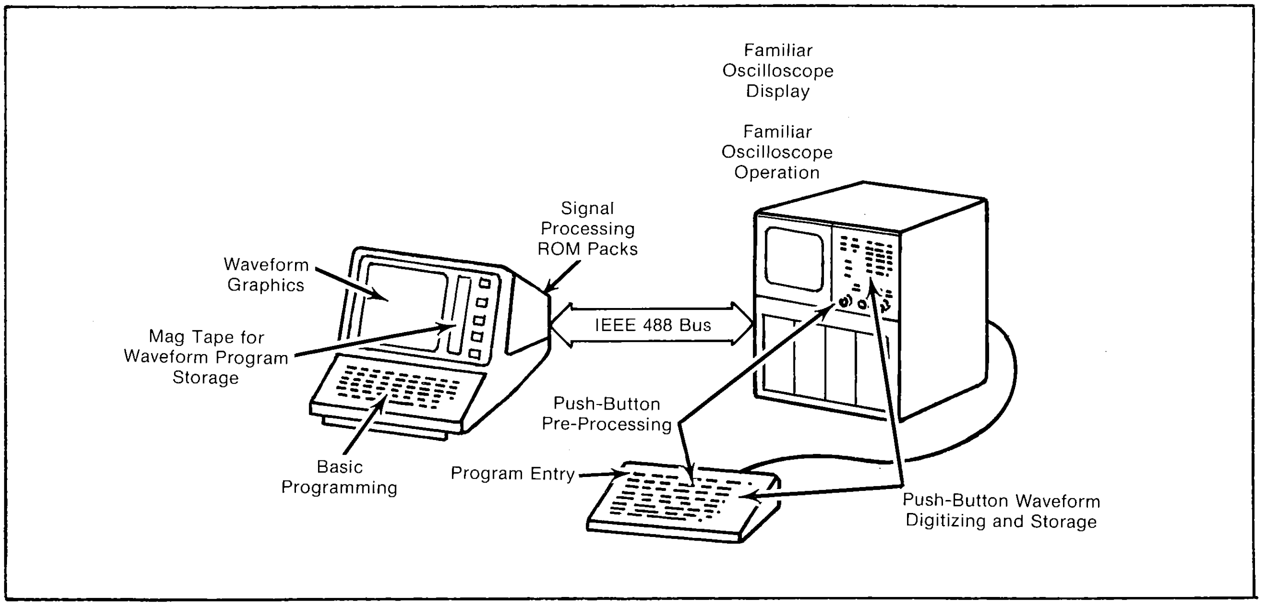



**Fig. 2.** *The WP1310 system components have a variety of features that combine for maximum benefits in waveform acquisition, processing, and storage.*

- Log and log-log displays
- Displays of system operating instructions
- Program listings and hard-copy capabilities

You can have them with the WP1310 system.

And there's still more. The WP1310 acquisition unit recognizes commands for changing the vertical and horizontal modes of mainframe operation. For example, you can send commands from the WP1310 controller over the bus to switch the acquisition unit from left vertical plug-in to right vertical plug-in. This allows you to have two different tests set up, one for each acquisition unit plug-in set, and to switch between them.

Of course the programs for doing these more complex processing and instrument control tasks can become lengthy. However, the WP1310 system controller has memory capabilities of up to 64 kilobytes with option 24. So there is plenty of room for program and waveform storage in the controller. Also, the WP1310 system controller has a tape drive which gives you access to an additional 300 kilobytes of magnetic tape storage. This latter feature is particularly important for permanent waveform and program storage since the WP1310 acquisition unit has volatile memory (the memory contents are lost if power is interrupted). The controller's magnetic tape will keep your data and programs secure.

## Oscilloscope simplicity, digital resolution

With the WP1310, you set up for waveform measurements just like you would with a general-purpose laboratory oscilloscope. But then, instead of looking at the CRT display of the waveform and counting divisions to measure time or amplitude, press one of the acquire buttons (AQR for repetitive signals or AQS for single shot). This causes the acquisition unit to digitize the waveform supplied to its input and store the amplitude values in memory. Once in digital memory, a variety of high-resolution waveform measurements can be made quickly, easily, and automatically—often by pressing a single button on the instrument. The instrument does the division counting and scaling for you. And it does it quite well.

Because of the acquisition unit's 10-bit digitizer, vertical waveform values can be resolved to 1 part in 1024. Or in terms of a waveform display, that means you'll have the ability to detect amplitude differences as small as 0.01 division (based on a full scale of five vertical divisions above and below the center line). That's vertical resolution!

For horizontal or time resolution, you have a number of choices. You can digitize at 128, 256, 512, or 1024 points equally spaced in time on the waveform. With 1024 points selected and using the fastest calibrated time-base sweep of 0.5 nanoseconds per division, you can detect time differences as small as five picoseconds. Or, for really fast requirements such as evaluating optical fibers, a 7S12 Sampler plug-in can be used with the acquisition unit. With the 7S12 at its fastest rate of 20 picoseconds per division and using 1024 points, the sample interval then becomes 0.2 picoseconds.

The only requirement for such high degrees of time resolution is that the acquired waveform be repetitive. This is necessary for the asynchronous sampling (equivalent-time sampling) of the acquisition unit to build up a full complement of 1024 amplitude samples over several horizontal sweeps.

Since sampling is asynchronous at a 3.5 microsecond rate, the emphasis of the WP1310 system is on acquiring repetitive waveforms. However, provisions have also been added for sequential sampling of low-speed transients. This provision is made via the internal clock of a 7B87 time base. The 7B87 clock is used to gate sequential real-time sampling when the AQS (acquire single sweep) button on the acquisition unit is used for waveform acquisition. In this single-sweep mode, full sequential sampling (128, 256, 512, or 1024 points) with a minimum sample interval of about four microseconds can be obtained. Or, with an external clock, it is possible to obtain a two-microsecond sample interval. And, as a further benefit of sequential real-time sampling, pre-triggering becomes possible (see Fig. 3).

Digitizing in either the repetitive or single-sweep mode is accompanied by storage of the digitial waveform values and vertical and horizontal scale factors in the acquisition unit's memory. With the full memory option (option 2D), there are eight kilowords available in the acquisition unit for storing waveforms, constants, and analysis programs. The number of waveforms that can be stored varies according to the points per waveform. In the eight kilowords, space is provided for storing five 1024-point waveforms, or ten 512-point waveforms, or twenty 256-point waveforms, or forty 128-point waveforms.

Once in memory, any of the waveforms can be called up again, along with its scale factor information, for display on the acquisition unit's CRT. Up to nine waveforms can be displayed at one time. Also provided along with waveform storage in the eight-kiloword memory option is room for 100 constant registers and 2000 program elements (line numbers or commands). These storage areas can also be accessed for display on the acquisition unit's CRT.

### Front-end processing power

Beyond just containing a waveform digitizer

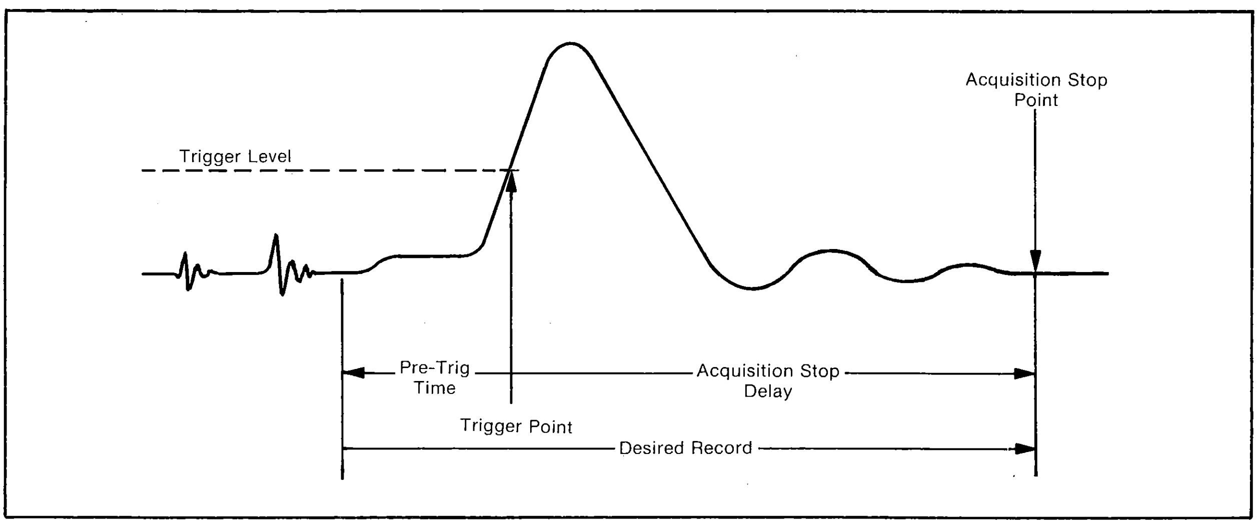



**Fig. 3.** *Pre-trigger: Unlike the repetitive mode where pseudorandom digitizing begins with the sweep trigger, the single-sweep acquisition mode using the 7B87 time base digitizes sequentially and continuously feeds samples to memory. This goes on until an "Acquisition Stop" freezes the most current samples in memory. By properly adjusting the "Acquisition Stop Delay" of the 7B87, those current samples can be made to include waveform data preceding the set trigger point. One benefit of this is that trigger levels can be set far above noise while still allowing capture of an entire leading edge in the pre-trigger zone.*

## WP1310 backs 400 MHz oscilloscope...

and memory for waveform storage, the WP1310 acquisition unit also has an internal microprocessor with firmware for controlling waveform acquisition and processing. Push buttons on the front panel and the Waveform Calculator keypad put a variety of measurement processes at your fingertips.

- Waveforms can be captured with signal averaging to improve signal-to-noise ratio (AVG).
- A variety of general parameters can be computed from stored waveforms (MAX, MIN, P-P, MEAN, MID, ENERGY, AREA and frequency or period).
- Pulse parameters can be determined (FALL, RISE, DELAY, WIDTH).
- And a variety of other computations and manipulations can be made.

All by simply pressing buttons, just like a calculator and the results are conveniently displayed on the oscilloscope CRT (Fig. 4).

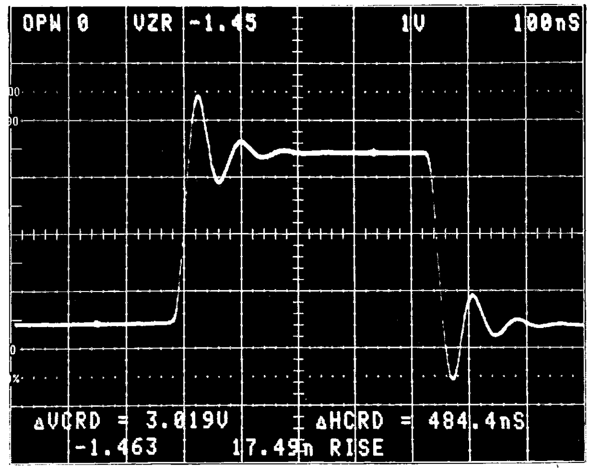
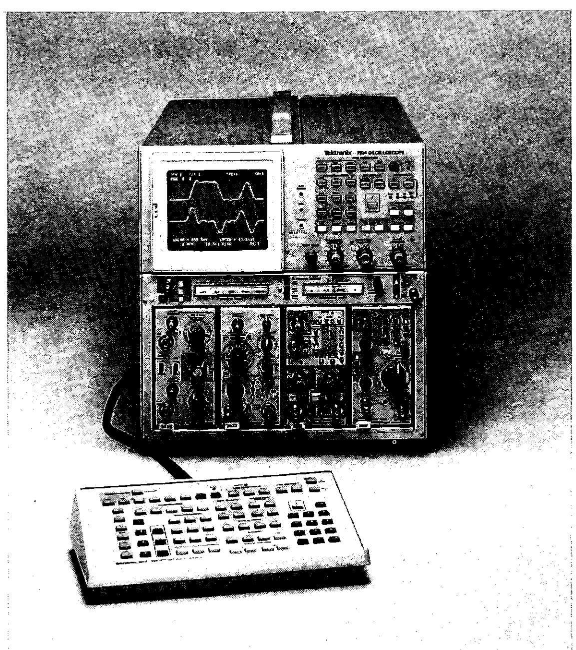



**Fig. 4.** *Stored waveforms can be recalled from WP1310 memory for redisplay at any time in a familiar oscilloscope fashion. Also, the displays are augmented by labels and waveform parameters computed by firmware.*

## The 7854 Oscilloscope, a closer look



The acquisition unit of the WP1310 Waveform Processing System is the new TEKTRONIX 7854 Oscilloscope. Based on the Tektronix line of high-performance 7000-Series oscilloscopes, the new 7854 Oscilloscope takes a significant step beyond standard oscilloscope capabilities. Waveforms acquired through its 400 MHz bandwidth mainframe can be viewed in real time or digitized in equivalent time and stored in digital memory for later viewing and analysis.

Waveform analysis is provided in the 7854 through visible screen cursors and push buttons that invoke internal microprocessor routines for computing a variety of waveform parameters and mathematical functions. There are buttons for:

| | |
|---|---|
| Delay time | Smooth |
| Pulse width | Integrate |
| Rise time | Differentiate |
| Fall time | Interpolate |
| Period | Recall ordinate |
| Frequency | Change ordinate |
| Maximum | Square root |
| Minimum | Natural log |
| Vertical midpoint | Exponential |
| Root mean square | Absolute value |
| Average value | Signum— +1, 0, or -1 |
| Area under curve | +, -, *, / |
| Energy | ., 0-9, and EEX |

These are some of the more important function keys. Plus, there are additional keys for program entry, program execution control, and waveform acquisition, positioning, display, and expansion. And there are still more capabilities and features—including GPIB (IEEE 488) compatibility.

For a complete description of the 7854 Oscilloscope, contact your local Tektronix Field Office. Or request WP1310 information, which includes a 7854 Oscilloscope data sheet, via the reply card bound into this issue of HANDSHAKE.

Being able to make a measurement at the touch of a button is a great time saver. Counting squares, multiplying by scale factors, and other finger, visual, mental, and graphic gymnastics are reduced or completely avoided. Minutes are saved on simple measurements. Hours can be saved on more complex operations.

Additionally, multi-stepped measurement sequences can be programmed from the Waveform Calculator keypad, stored in the acquisition unit's memory, and recalled for execution. This means that even more time can be saved. Instead of re-pushing many buttons in sequence to perform each iteration of an analysis series, the buttons can be pushed once in the PROGRAM ENTRY mode. Then the entire sequence can be run automatically, as often as you like, in the EXECUTE mode. Not only does the button sequence run faster as a program, but measurement errors associated with manually executed sequences are dramatically reduced. Programs execute button functions the same every time—they don't forget steps or inadvertantly execute button functions out of the established sequence. That means a dramatic increase in measurement repeatability.

### IEEE 488 compatibility

Once a waveform has been acquired and pre-processed by the WP1310 acquisition unit, it can be transferred to the WP1310 system controller for high-level processing. The transfer is over an IEEE 488 bus.

The same bus, because of IEEE 488 compatibility, can be used to tie more acquisition units into the WP1310 Waveform Processing System. You could have the WP1310 controlling up to 14 acquisition units in a multiple station test area for example. The programs could be down loaded to each acquisition unit according to the test needs at each station. With the WP1310 processing distributed between the system controller and the acquisition unit...well, you have just the measurement power you need when you need it and where you need it.

*By Bob Ramirez*
*HANDSHAKE Staff*

# New TEK SPS BASIC releases offer more capabilities, more memory

Two new releases of TEK SPS BASIC software, V02-02 and V02XM-02, are now available for updating your signal processing system. Both releases retain the previous capabilities of TEK SPS BASIC as well as offer the additional capabilities of

- A high-level GPIB driver
- Support for additional peripherals
- Additional programming convenience
- A 7612D commands package

And, beyond these features, V02XM-02 offers extended memory for handling large waveform arrays such as those that can come from the new 7612D Programmable Digitizer or many smaller arrays such as might come from any other digitizer.

The importance of these new capabilities becomes apparent in taking a closer look at each of them. For owners of the new 7612D Programmable Digitizer, the 7612D commands package and the extended memory version will be of particular interest.

### High-level GPIB driver

The low-level GPIB driver is still a part of the TEK SPS BASIC monitor. And it still offers the greatest flexibility in dealing with a wide variety of interpretations and implementations of the GPIB standard.

The new high-level GPIB driver, however, can make life a lot easier when dealing with certin standardized data transfers over the GPIB. It reduces the number of program lines necessary for communicating with many GPIB instruments, and the commands are simpler and easier to remember. With the high-level GPIB driver, a single command is all that is necessary now to acquire data from either the 7612D or 7912AD Programmable Digitizers.

## New TEK BASIC releases...

### Additional peripheral support

If you would like to add either an RL01 disk drive or an RX02 double-density floppy disk drive to your system, the new releases of TEK SPS BASIC software have been augmented to support these devices. This is in addition to the original line of peripherals supported by the previous software releases.

Also, both new releases have an enhanced graphics keyboard driver. This lets you exploit the higher resolution of TEKTRONIX 4014 Computer Display Terminals fitted with the Enhanced Graphics Module option. With the enhanced graphics, your signal processing data can be automatically displayed with the resolution of a 0-3114 Y and 0-4095 X coordinate system. And you won't have to modify your existing TEK SPS BASIC graphics programs. They'll run with the enhanced graphics.

### Greater programming convenience

For greater convenience in programming, several new commands have been added to the new releases of TEK SPS BASIC. For example, there is now a HASH command which assists recorded I/O file access. Also, an LST command produces program listings with indented FOR/NEXT loops and clearer separation of multiple statement lines. And terminal control characters can now be used to suppress or resume output to a terminal.

Those are just a few examples of the additions for your programming convenience. Many other commands and drivers have also been extended with options not included in previous release of the software.

### 7612D commands added

Two commands have been added to support signal acquisition with the new 7612D Programmable Digitizer. One command is for signal averaging, and the other is for fast logging of 7612D data to peripheral storage. With these two new commands, it is possible to handle as many as one hundred 256-point waveforms a second from a 7612D Programmable Digitizer installed in a WP3201 system.

### More waveform storage

The extended memory version of TEK SPS BASIC software (V02XM-02) is designed to handle a higher volume of waveform data than the previous version. Intended for use with a Digital Equipment Corporation PDP-11 minicomputer with KT11D Memory Management hardware, this extended memory version of TEK SPS BASIC adds 96 kilowords of memory for storage of floating-point and integer arrays. That means you now can have a total of 124 kilowords of user-available memory—the standard 28 K for software and program storage and 96 K for waveform arrays. For very much processing of large arrays, such as can come from a 7612D Programmable Digitizer, this additional memory is a necessity.

### And it's all compatible

Perhaps the best news is that both of the new releases of TEK SPS BASIC are designed to run programs written on the previous release. But then you'll probably still want to do some modification of older programs to take advantage of the new features in TEK SPS BASIC V02-02 and TEK SPS BASIC V02XM-02.

Tektronix® COMMITTED TO EXCELLENCE

HANDSHAKE
Newsletter of the Programmable Instruments Users Group



**HANDSHAKE**
**Group 157 (94-384)**
**Tektronix, Inc.**
**P.O. Box 500**
**Beaverton, Oregon 97077**

A-4463

Tektronix®
COMMITTED TO EXCELLENCE

# HANDSHAKE

# APPLICATIONS LIBRARY

PROGRAM 5-40.
**Dynamic Performance Testing of Digitizers**

Instruments Required: None, but stored instrument data is required.
Software Packages Required: Graphics and Signal Processing.
Listing Length: 1620 lines.

This analysis sequence uses two programs to determine digitizer dynamic performance. The first program in the sequence operates on stored output data from the digitizer to synthesize the digitizer's analog input. The synthesis is done by least-squares fitting an appropriate model to the digitizer output data. Then, key statistics describing the digitizer's dynamic performance are computed and stored. The second program provides several options for further analyses of the stored performance statistics.

**Additional Information**

"Dynamic testing reveals overall digitizer performance," DeWitt, Laurie. HANDSHAKE, Tektronix, Inc., Vol 5, No 1, Spring/Summer 1980, pg 9.

**PROGRAM NAMES:** MDPMPN, MDPANL

**AUTHORS:** Lyle Ochs, Paul McClellan
Signal Analysis Group
94-384

**DATE:** November 10, 1977

**ABSTRACT:**

MDPMPN and MDPANL are TEK SPS BASIC programs used to digitally measure the dynamic performance of digitizers. Program MDPMPN first synthesizes a digitizer's analog input by least squares fitting an appropriate model to the digitizer output. It then calculates summary statistics describing the performance of the digitizer and, if desired, stores data for later analysis by program MDPANL. This program supports a variety of analysis options that may be applied to the digitizer data and other data generated by MDPMPN.

**DISCUSSION:**

Historically, A/D and digitizer specifications have been of the DC or static type and have often been unclear or incomplete. It is sometimes desirable to augment these static specifications with actual measurements of digitizer performance.

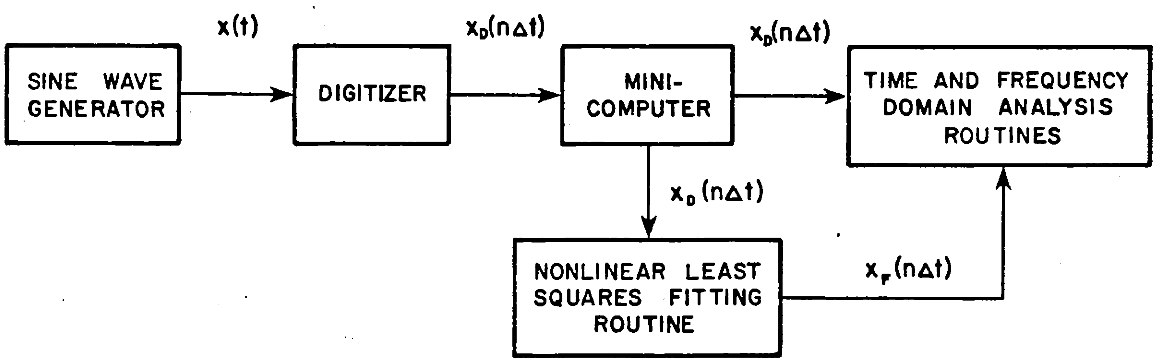
In the past, test setups for measuring digitizer performance have involved specialized analog instrumentation and reference signals. However, valid performance specifications result only if the errors contributed by the instrumentation are small compared to those of the A/D under test. This can be an especially troublesome problem in testing the higher resolution, higher speed A/D converters available today.

A direct procedure for evaluating digitizer performance is to interface the instrument to a computer and analyze the output in its own domain -- digitally. In this procedure, the test setup consists of a signal generator, digitizer, minicomputer, graphics terminal, and a set of analysis routines. Typical choices for the signal generator include precision voltage sources for static testing and sine wave generators or random noise generators for dynamic tests.

In this digital approach, there are no extraneous error components from D/A's, comparators, scopes, etc. found in test setups employing analog instrumentation for measurement and analysis. But specifications for the reference signal are critical in this direct, digital test setup, and they must not exceed theoretical accuracy, resolution, and noise limits of the digitizer to be tested. For sine wave testing, the insertion of a bandpass filter between the signal generator and digitizer can substantially reduce these problems by rejecting harmonic distortion components and reducing noise bandwidth.

Focusing on dynamic testing with sine waves for reference signals, a time domain approach can be taken to analyze digitizer performance. This method involves synthesizing the analog input (reference) signal by fitting a sine wave model to the digitizer output via nonlinear least squares. It allows for both time and frequency domain analysis.

Programs MDPMPN and MDPANL have been developed to analyze the digitizer output data using this time domain approach. Program MDPMPN synthesizes the analog input and calculates some measures of digitizer performance. If desired, it will also save data for further analysis by program MDPANL. The diagram outlines the technique.




In this approach, the analog input signal is a sine wave represented mathematically by the equation:

$$X(t) = A \sin (2\pi ft+\Theta) + C,$$

where A, f, $\Theta$, and C are the amplitude, frequency, phase, and DC offset parameters. The digitizer output is $X_D$ ($n\Delta t$), where $\Delta t$ is the sampling interval, and n is the sample index. MDPMPN requires that the digitizer output be stored in a data file. The program reads the data and overlays a subroutine to synthesize the analog input. In this subroutine the digitizer output and the sine wave frequency are first used to form initial estimates of the remaining parameters A, $\Theta$, and C. All this data is then used in a nonlinear least squares fitting routine which determines the fitted parameter values.

MDPMPN next overlays a subroutine to calculate summary statistics. The model with fitted parameters is:

$$X_F(t) = \hat{A} \sin (2\pi\hat{f}t+\hat{\Theta}) + \hat{C}.$$

This is taken to be the analog input to the digitizer. $X_F(t)$ is "perfectly sampled" by computer subroutine, producing $X_F(n\Delta t)$, and this perfectly sampled analog input is subtracted, sample by sample, from the digitizer output to form the analog error signal

$$r(n\Delta t) = X_D(n\Delta t) - X_F(n\Delta t).$$

The minimum, maximum, RMS analog error, and signal-to-noise ratio are calculated and printed. The signal-to-noise ratio is the ratio of RMS synthesized input (with mean removed) to RMS analog error, expressed in dB's.

The summary statistics subroutine also "perfectly quantizes" the sampled input data $X_F(n\Delta t)$ -- call this data $X_{FD}(n\Delta t)$ -- and the analog error signal for an ideal digitizer with the same number of bits as the digitizer under test is calculated. In equation form, the errors are

$$R(n\Delta t) = X_{FD}(n\Delta t) - X_F(n\Delta t).$$

The ratio RMS(r)/RMS(R) provides a measure of actual digitizer performance relative to ideal performance. Taking the base two logarithm gives "lost bits". For a b bit digitizer, the effective bits for the digitized data $X_D$ is then

$$EB(X_D) = b - \log_2(RMS(r)/RMS(R)),$$

which is also calculated and printed. Note that effective bits may decrease due to either systematic or random errors in the digitization process since either type of error will tend to inflate RMS(r).

The above least squares fitting procedure and summary statistics calculations are automatically made by program MDPMPN. If further performance measurements are desired, the program will save the digitized output, synthesized input, analog errors, ideally digitized output, and ideal digitizer analog errors on data files for further analysis by program MDPANL. The analysis options available are 1) graphing one or two data sets, 2) computing and graphing amplitude spectra of one or two data sets, 3) estimating RMS time jitter and RMS residual additive error for sine wave input signals, 4) calculating and graphing RMS analog error versus digital code, and 5) calculating and graphing RMS analog error versus phase of sine wave input signals.

Time jitter is defined for our purposes as the time uncertainty in the points at which the input signal was sampled and quantized. The signal at the time jittered point $n\Delta t+e_n$ may be approximated by the first order Taylor series

$$X(n\Delta t+e_n) \doteq X(n\Delta t) + e_n X'(n\Delta t).$$

The amplitude uncertainty due to the time jitter $e_n$ is then

$$X(n\Delta t+e_n) - X(n\Delta t) = e_n X'(n\Delta t),$$

so the effect of time jitter in the digitizer output is an amplitude uncertainty that is proportional to the slope of the input signal at each sample point. This uncertainty is in addition to the digitizer's quantization errors, which are independent of the slope of the input signal (except when coherently sampling). Hence to detect and estimate time jitter from digitizer output, slope dependent analog errors need to be detected and measured.

To do this, the following mean square analog error model is used:

$$E_t^2(s^2) = E_r^2 + E_j^2\, s^2,$$

where $E_t^2(s^2)$ is the total mean square analog error as a function of squared input slope, $s^2$, $E_j^2$ is the mean square time jitter, and $E_r^2$ is the residual mean square analog error, independent of the input slope s. The input slope is calculated and sampled using the fitted input signal parameters:

$$s(n\Delta t) = X_F'(n\Delta t) = 2\pi\hat{A}\hat{f}\sin(2\pi\hat{f}n\Delta t + \hat{\Theta})$$

For sine wave inputs, the range of $s^2$ values is divided into intervals. The analog errors, $r(n\Delta t)$, are grouped according to these $s^2$ intervals, and for each interval the mean square analog error, $E^2{}_t\,(s^2)$, is calculated. In this way, the independent and dependent variables for fitting a straight line (simple linear regression) are constructed. For some values of $s^2$ there may be no observations, and these $s^2$ values are then ignored in the remaining calculations. $E_r^2$ and $E_J^2$ are then estimated by fitting a straight line to this data by least squares, giving $\hat{E}_r^2$ and $\hat{E}_J^2$. If $\hat{E}_J^2 \leq 0$, then time jitter has not been detected and we set $\hat{E}_J^2=0$, $\hat{E}_r^2$ = mean square analog error. Otherwise, $\hat{E}_J^2$ is tested for statistical significance and RMS time jitter calculated. $\hat{E}_r^2$ is also tested for significance and RMS residual error calculated.

The above analog error model is valid only for a range of time jitter values, depending on the sine wave input used. If $E_J^2$ is too small, the amplitude effects of time jitter will be buried in quantization noise. If it is too large, additional, higher order terms would need to be included in the error model. For an input sine wave signal described by $A\sin(2\pi ft + \Theta) + C$, where A is expressed in LSBs, the interval over which the model is valid is:

$$1/(2\pi A) < E_j f < 1/(\pi\sqrt{2A})$$

MDPANL calculates this interval along with the above estimates.

It is important to keep in mind that the time jitter estimation procedure does not directly measure time jitter. Rather, it looks for a component of mean square analog error that is linearly related to the squared input slope in a positive sense. If this dependence is detected, the amount of time jitter that could have produced the same amount of linear dependence is then estimated. If the time jitter lies within the interval defined above, the time jitter should be detected in this way. However, if systematic errors occur, especially if they are near the sine wave "zero" crossings, this procedure may detect significant "time jitter" when none or an insignificant amount is really present. Thus the time jitter estimate should not be viewed in isolation from a graph of the analog errors and other analyses.

By systematically varying the frequency of the input sine wave signal and repeating some or all of the analyses provided by programs MDPMPN and MDPANL for each frequency, dynamic performance versus frequency can be graphically displayed.

The above programs may also use ramp or DC input signals to help evaluate digitizer performance. Ramp signals can be used to measure digitizer dynamic linearity at varying ramp rates. Some digitizers may have static nonlinearities across their record lengths that can be most easily seen in acquired DC signals. However, not all of the analyses performed on digitized sine wave data can be performed on digitized ramp or DC data. Also, it is generally more difficult to evaluate the accuracy of a ramp signal generator than a sine wave signal generator.

Programs MDPMPN and MDPANL can be used in measuring the dynamic performance of digitizers. However, static, or DC, performance measurements may also be desired. These measurements can be made using program ADSPAN, documented elsewhere.

## OPERATING ENVIRONMENT AND INSTRUCTIONS:

### A. Signal Generator Requirements

Any sine wave, ramp, or DC generator used to supply test signals should generate signals accurate to within the resolution of the ideally digitized signals.

A candidate sine wave generator can be evaluated using a spectrum analyzer. The total harmonic distortion and noise level should be less than the total harmonic distortion of a sine wave digitized by an ideal digitizer with the same number of bits as the digitizer under evaluation. For an ideal b-bit quantizer, the total harmonic distortion is $THD \doteq - 6.02\ b - 1.76$ dB's. If the spectrum analyzer does not have sufficient dynamic range to verify that the signal generator satisfies this requirement, a bandpass filter can be inserted between the signal generator and digitizer to reduce what harmonic distortion is present to be sure it is satisfied. This will also reduce the noise bandwidth.

Also, the frequency of the sine wave generator output should be stable over the digitized record length. Frequency jitter can be detected on a spectrum analyzer as a widening of the fundamental peak. For a given sine wave frequency f, the allowable short-term frequency jitter $\Delta f$ is

$$|\Delta f/f| \leq e/(2^b \pi n),$$

where n is the number of cycles in the record to be digitized, b is the number of bits in the digitizer output words, and e is the acceptable amplitude uncertainty in LSBs. A reasonable choice for e is 1/4 LSB. For example, suppose 100 cycles of a 20 megahertz sine wave is to be acquired on an 8-bit digitizer, and e = 1/4. Then $|\Delta f| \leq 62.2$ hertz. If e is expressed as a RMS value, $\Delta f$ is also. For lower frequencies, many cycles, and many bits, this may be a difficult performance requirement to verify.

Ramp signal generators can be difficult to evaluate. A candidate ramp generator's output should be linear to well within the resolution of the digitizer under test, say 1/4 LSB or less. If another digitzer with greater resolution than the one under evaluation is available (say having two or more additional bits) the ramp signal could be digitized with it and its performance verified. Otherwise, the signal might be passed through a differentiator and the result evaluated as an ideal DC signal. However, the differentiator may have its own nonlinearities, distorting the measurement.

A DC signal might also be evaluated by digitizing it with a digitizer with higher resolution than the digitizer under evaluation. Or its AC component may be viewed on a sufficiently sensitive oscilloscope. However, in either of these methods the nonlinearities of the measuring instrument should be adjusted for.

## B. Program Operating Instructions.

Data acquisition and analysis is divided into three conceptual phases, each requiring a BASIC program to be OLDed and RUN.

**1) Data Acquisition** (user supplied program). This is the transfer of data from the local memory of the digitizer under test to an ASCII file required by the "MDP" programs. The data should be previewed at this time and the ASCII file must have a form defined by the following PRINT statements:

```
PRINT #1,ID$
PRINT #1,ST$
PRINT #1,NB,DT,F,ND,IB,IE
FOR I=0 TO ND-1
PRINT #1,Y(I)
NEXT I
```

where:

ID$--An identification string for the digitizer data, e.g., date, instrument, etc.

ST$--A string designating the signal type. ST$ may be DC, RAMP, or SINE currently. These are the only signal types that the MDP routines know how to fit.

NB---The number of bits in the digitizer words.

DT---The sampling interval in seconds.

F----If the signal type is SINE, this variable should be set to its frequency. For other signal types its value is ignored.

ND---The number of acquired data points, Y(0),Y(1),...,Y(ND-1).

IB,IE--The starting and ending indices of the data to be analyzed. Typically IB=0 and IE=ND-1.

Y(0),Y(1),...,Y(ND-1)--The ND NB-bit digitizer words. Y should be an integer array.

The TEK SPS BASIC routine to transfer digitizer output from local memory to computer memory, preview the data, and write out the raw data file is provided by the user.

**2) Fitting Data, Generating Summary Statistics and Intermediate Data Files** (program MDPMPN). Once the raw data is on file, a fitting routine can be called to construct a computer synthesized version of the analog signal that was input to the digitizer. Summary statistics are then printed and intermediate data files can be created for later and more complete analysis.

To complete the above steps, simply OLD the program on file MDPMPN.BAS and RUN it. It first asks for the raw data file name, reads the information on it, and graphs the data points Y(IB),Y(IB+1),...,Y(IE) to be analyzed. The program will next ask for any desired changes in IB and IE and then overlay the appropriate least squares fitting routine (MDPRMP.BNR for DC or RAMP signals or MDPSIN.BNR for SINE signals). After the analog input has been synthesized, the program finally overlays and calls MDPSS.BNR. This routine asks for a file name for output of intermediate data, if desired, then computes and prints summary statistics, and, finally, writes out the intermediate data if that option was chosen.

The main program, MDPMPN.BAS, can be immediately rerun on the same or a new raw data file, unless a data record longer than 1024 points was analyzed. In that case, parts of the main program were deleted to provide needed additional memory space and MDPMPN.BAS will need to be OLDed again.

**3) Analysis of Intermediate Data** (program MDPANL). By OLDing and RUNning MDPANL.BAS, various analysis routines can be overlaid and the intermediate data read in and analyzed. MDPANL.BAS prints a "menu" of analysis options and data sets. Analysis routines are named MDPAXX.BNR, where XX is a one or two digit integer. The ".BNR" extension is used to designate a BASIC routine stripped of REMark statements. Currently five analysis options are available. These are:

MDPA1.BNR--graphs one or two data sets.

MDPA2.BNR--computes and graphs amplitude spectra of one or two data sets.

MDPA3.BNR--estimates RMS time jitter and RMS residual additive error for SINE type input signals.

MDPA4.BNR--calculates and graphs RMS analog error versus digital output code.

MDPA5.BNR--calculates and graphs RMS analog error versus phase of SINE type input signals.

Note that options 3 and 5 require SINE type input signals. Only analysis options 1 and 2 are valid for DC type signals, and only 1, 2, and 4 are valid for RAMP type signals.

Additional analysis routines can be written by the user; just follow the data reading, program line number, and house cleaning conventions of the above routines.

## C. Additional Program Notes:

1) If analysis option MDPA2.BNR or any other analysis using the FFT is to be used, the length of the data record to be analyzed, IE-IB+1, must be a power of 2.

2) A number of control parameters are set in the nonlinear least squares fitting routine used for SINE type signals (MDPSIN.BNR). For an explanation of these, a copy of MDPSIN.BAS (with REMarks) is available.

3) If a sawtooth signal is used to generate RAMP type input signals, only one ramp at a time should be fitted by MDPRMP.BNR.

4) Less than one cycle of a SINE type signal can be fitted by MDPSIN.BNR if the data points are not clustered around an extremum.

5) The units of the digitizer output are set to seconds ("S") and least significant bits ("LSB").

6) To speed up data storage and retrieval from raw data files and also to reduce raw data file lengths, replace PRINT's and INPUT's by WRITE's and READ's, respectively, when storing or retrieving raw data in the user supplied program and program MDPMPN.

7) The program files necessary for phases **2)** and **3)** above are:

MDPMPN.BAS
- MDPRMP.BNR
- MDPSIN.BNR
- MDPSS.BNR

MDPANL.BAS
- MENU.MDP (menu file, not a program)
- MDPA1.BNR
- MDPA2.BNR
- MDPA3.BNR
- MDPA4.BNR
- MDPA5.BNR
- .
- .
- .

Copies of the above routines with remarks (.BAS extension) are available.

**PROGRAM 5-40.**
**Dynamic Performance Testing of Digitizers**

## Program Listings

The following listings are included as part of this program:

| NAME | DESCRIPTION |
|---|---|
| MDPMPN.BAS | Main program for measuring digitizer performance. |
| MDPRMP.BAS | Routine for fitting DC or ramp type signals. |
| MDPSIN.BAS | Routine for fitting sine waves. |
| MDPSS.BAS | Routine to compute summary statistics and write data files. |
| MDPANL.BAS | Routine for overlaying. |
| MDPA1.BAS | Routine to input and graph up to two data sets. |
| MDPA2.BAS | Routine to compute and graph amplitude spectra of one or two sets of data from disk. |
| MDPA3.BAS | Routine to estimate and display RMS time jitter and RMS residual quantization error from the analog errors of a digitized sinusoidal signal. |
| MDPA4.BAS | Routine to calculate and graph RMS analog error versus digitizer output code. |
| MDPA5.BAS | Routine to calculate and graph digitizer RMS analog error versus phase. |

(MDPMPN.BAS) PROGRAM 5-40. DYNAMIC PERFORMANCE TESTING OF DIGITIZERS

```
100 REM **** MDPMPN.BAS  MEASURING DIGITIZER PERFORMANCE: MAIN       ****
105 REM **** PROGRAM, FOR FITTING DIGITIZER OUTPUT DATA AND POSSIBLY ****
110 REM **** WRITING VARIOUS DATA FILES ON THE SYSTEM PERIPHERAL     ****
115 REM ****                TEK SPS BASIC, NOVEMBER 10, 1977         ****
120 REM
125 REM GET DATA FILE NAME, READ IN HEADER INFO AND DATA
130 PAGE
135 PRINT "DATA FILE NAME";
140 INPUT FN$
145 CLOSE #5
150 OPEN #5 AS FN$ FOR READ
155 INPUT #5,ID$,ST$
160 ST$=TRM(ST$)
165 INPUT #5,NB,DT,F,ND,IB,IE
170 REM
175 DELETE Y,WY
180 INTEGER Y(ND-1)
185 HU$="S"\VU$="LSB"
190 WAVEFORM WY IS Y,DT,HU$,VU$
195 REM
200 FOR I=0 TO ND-1
205 INPUT #5,Y(I)
210 NEXT I
215 CLOSE #5
220 REM
225 REM GRAPH DATA AND ASK FOR ANY CHANGES IN
230 REM DATA INTERVAL TO BE FIT
235 PAGE
240 PRINT "I.D.: ";ID$;"    SIGNAL TYPE: ";ST$
245 PRINT "ND = ";ND;"  IB = ";IB;"  IE = ";IE
250 GRAPH Y(IB:IE)
255 RELEASE "GRAPH"
260 PRINT "^G^G"
265 WAIT
270 PAGE
275 PRINT "DO YOU WISH TO CHANGE IB OR IE?"
280 INPUT E$
285 IF E$='Y' THEN E$='YES'
290 IF E$<>'YES' THEN 340
295 PRINT "IB ="
300 INPUT IB
305 PRINT "IE ="
```

```
310 INPUT IE
315 IF IE<=IB THEN 295
320 GOTO 235
325 REM
330 REM FETCH APPROPRIATE OVERLAY TO FIT TYPE OF DATA JUST
335 REM READ IN (DC,RAMP,SINE)
340 DELETE 2500,4999
345 IF ST$="DC" THEN OVERLAY "MDPRMP.BNR"
350 IF ST$="RAMP" THEN OVERLAY "MDPRMP.BNR"
355 IF ST$="SINE" THEN OVERLAY "MDPSIN.BNR"
360 REM
365 REM SET UP ARRAYS, VARIOUS CONTROL PARAMETERS FOR FITTING
370 REM ROUTINE AND PRINT PRELIMINARY INFO ON KEYBOARD
375 GOSUB 2520
380 DELETE 2500,2999
385 REM
390 REM FIT THE DATA
395 GOSUB 3067
400 REM
405 REM PRINT FINAL PARAMETER ESTIMATES, STANDARD ERRORS, A FEW STATS
410 GOSUB 4020
415 REM
420 REM DELETE SUBS BELOW, FETCH OVERLAY TO COMPUTE SUMMARY STATISTICS,
425 REM POSSIBLY CREATE VARIOUS DATA FILES FOR LATER ANALYSIS
427 DELETE 3000,4000
430 DELETE 3000,4999
435 IF N>1024 THEN DELETE 100,425
440 OVERLAY "MDPSS.BNR"
445 GOSUB 2550
450 END
455 REM **** END MAIN MDP PROGRAM ****
460 REM
```

```
2500 REM ***************   MDPRMP.BAS          **********************
2502 REM **** ROUTINE FOR INITIAL PRINT-OUT FOR FITTING DC OR *****
2504 REM **** RAMP TYPE SIGNALS                                 *******
2505 REM
2506 REM INPUTS: IB,IE,ID$,SN$,FN$,NB,DT,F,ND
2508 REM
2515 REM DEFINE DATA LENGTH, PRINT HEADER INFO
2520 N=IE-IB+1
2525 PAGE
2530 PRINT "I.D.";ID$
2535 PRINT "SIGNAL TYPE:  ";ST$
2540 PRINT "DATA FILE NAME:  ";FN$
2545 PRINT
2550 PRINT "NB=";NB;"     DT=";DT
2555 PRINT "ND=";ND;"   IB=";IB;"   IE=";IE
2560 PRINT
2565 REM
2570 RETURN
2575 REM **** END INTIAL PRINT-OUT ROUTINE FOR DC,RAMP TYPE SIGNALS  ****
2580 REM
3000 REM **** ROUTINE FOR SIMPLE LINEAR REGRESSION  *****
3002 REM *****    OR ESTIMATING DC          ***********
3005 REM
3010 REM INPUTS: IB,IE,N,Y,DT
3015 REM OUTPUTS:
3016 REM FOR DC TYPE SIGNALS FIND:
3018 REM MS--MEAN OF Y(IB) THRU Y(IE)
3020 REM SE--STANDARD ERROR OF THE MEAN
3022 REM CS--RESIDUAL SUM OF SQUARES
3024 REM R2--R-SQUARED VALUE
3026 REM
3028 REM FOR RAMP TYPE SIGNALS FIND:
3030 REM MS--MEAN OF Y(IB) THRU Y(IE)
3032 REM SS--RESIDUAL SUM OF SQUARES
3034 REM CS--CORRECTED SUM OF SQUARES
3036 REM AH--ESTIMATE OF THE INTERCEPT
3038 REM BH--ESTIMATE OF SLOPE
3040 REM AE--STANDARD ERROR FOR AH
3042 REM BE--STANDARD ERROR FOR BH
3045 REM
3050 REM
3055 REM OVERWRITTEN VARIABLES:MS,SY,MT,ST,BH,SS,CS,YT,CT,TT,I,RS,R2,SE
```

```
3056 REM                        AND S.
3060 REM
3067 IF ST$="DC" THEN GOTO 3205
3068 REM
3070 REM  FOR RAMP TYPE SIGNALS
3072 MS=MEA(Y(IB:IE))
3074 SY=N*MS
3076 MT=DT*(IB+IE)/2
3080 REM
3085 BH=0\CS=0\CT=0\TT=0
3090 FOR I=IB TO IE
3092 T1=DT*I
3094 T2=T1-MT
3096 T3=Y(I)-MS
3100 CS=CS+T3*T3
3105 TT=TT+T1*T1
3110 BH=BH+T2*T3
3120 CT=CT+T2*T2
3125 NEXT I
3130 REM THE PARAMETER ESTIMATES:
3135 BH=BH/CT
3140 AH=MS-BH*MT
3145 REM
3146 REM COMPUTE RESIDUAL SUM OF SQUARES, EST. NOISE VARIANCE
3148 RS=0
3150 FOR I=IB TO IE
3152 T1=Y(I)-(BH*DT*I+AH)
3154 RS=RS+T1*T1
3156 NEXT I
3158 REM
3160 S2=RS/(N-2)
3162 R2=1-RS/CS
3165 REM
3170 REM THE PARAMETER STANDARD ERRORS:
3175 BE=SQR(S2/CT)
3180 AE=SQR(S2*TT/(N*CT))
3185 REM
3190 GOTO 3245
3195 REM
3200 REM  FOR DC TYPE SIGNALS
3205 MS=MEA(Y(IB:IE))
3210 RS=RMS(Y(IB:IE))
```

```
3215 RS=N*RS*RS
3220 CS=RS-N*MS*MS
3225 S=SQR(CS/(N-1))
3230 SE=S/SQR(N)
3235 R2=1-CS/RS
3240 REM
3245 RETURN
3250 REM  ***** END FIT ROUTINE FOR DC,RAMP TYPE SIGNALS   *****
3255 REM
4000 REM ****  ROUTINE TO PRINT OUT FINAL RESULTS OF FIT  ****
4005 REM
4010 REM INPUTS FOR
4012 REM DC SIGNAL: MS,SE,R2,SN$
4014 REM RAMP SIGNAL: AH,AE,BH,BE,R2,SN$
4015 REM
4020 IF ST$="DC" THEN GOTO 4055
4022 PRINT
4025 PRINT "PARAMETER ESTIMATES:  SLOPE=";BH;"  INTERCEPT=";AH
4027 PRINT "STANDARD ERRORS:           ";BE;"           ";AE
4030 PRINT
4035 PRINT "R^2=";R2
4040 GOTO 4075
4045 REM
4050 REM PRINT-OUT FOR DC TYPE SIGNALS
4055 PRINT
4060 PRINT "ESTIMATED MEAN=";MS;"  STANDARD ERROR OF THE MEAN=";SE
4065 PRINT "R^2 VALUE=";R2
4070 REM
4075 RETURN
4080 REM **** END FINAL PRINTOUT FOR DC,RAMP FIT ROUTINES  ****
4085 REM
```

```
2500 REM **************       MDPSIN.BAS        ********************
2502 REM **** ROUTINE TO SET UP FOR SINE WAVE FIT ROUTINE   *****
2504 REM
2506 REM INPUTS: FN$,ID$,ST$,NB,DT,F,ND,IB,IE
2508 REM OUTPUTS: P,PA,PE,ES,NI,IO,N (SEE BELOW FOR DEFINITIONS)
2510 REM OVERWRITTEN VARIABLES: P,PE,PA,ES,NI,IO,N
2512 REM
2515 REM SET UP ARRAYS, CONTROL PARAMETERS
2520 P=4
2525 DELETE PA,PE
2530 DIM PA(P-1),PE(P-1)
2535 ES=2E-05
2540 PE=ES
2542 NI=8
2545 IO=1
2546 PA(1)=F
2548 N=IE-IB+1
2550 KT=ITP(N*DT*PA(1))
2552 IF KT<1 THEN GOSUB 2620
2554 IF KT>=1 THEN GOSUB 2724
2565 REM
2570 REM PRINT HEADER INFO FROM USER'S DATA FILE
2572 PAGE
2574 PRINT "I.D.";ID$
2576 PRINT "SIGNAL TYPE:";ST$
2578 PRINT "DATA FILE NAME:";FN$
2580 PRINT
2582 PRINT "NB=";NB;"   DT=";DT;"   F=";F
2584 PRINT "ND=";ND;"   IB=";IB;"   IE=";IE
2586 PRINT
2588 PRINT "INITIAL PARAMETER ESTIMATES:"
2590 PRINT "AMP=";PA(0);"  FREQ=";PA(1);"  PHASE=";PA(2);"  DC=";PA(3)
2592 PRINT
2594 RETURN
2596 REM **** END SET-UP FOR NON-LINEAR LEAST SQUARES-FIT ***
2598 REM
2600 REM **** ROUTINE TO FIND INITIAL PARAMETER ESTIMATES FOR ****
2602 REM **** NONLINEAR LEAST SQUARES SINE WAVE FIT TO DATA   ****
2604 REM **** CONTAINING LESS THAN ONE CYCLE.                ****
2606 REM
2608 REM INPUTS:  IB,IE,N,DT,PA(1),Y
2610 REM OUTPUTS:  PA,CS
```

```
2612 REM NEW OR OVERWRITTEN VARIABLES: MS,CS,Q,TP,NS,DI,I,AR,SN,CO,PA,PO
2614 REM
2616 REM GET CORRECTED SUM OF SQUARES
2618 REM
2620 MS=MEA(Y(IB:IE))
2622 CS=RMS(Y(IB:IE))
2624 CS=N*(CS*CS-MS*MS)
2626 REM
2628 REM FIT SINE+COSINE MODEL TO AT MOST 50 DATA POINTS USING USER
2630 REM PROVIDED FREQUENCY ESTIMATE, PA(1), AND LINEAR LEAST SQUARES.
2632 REM ASSEMBLE AUGMENTED COEFFICIENT MATRIX.
2634 REM
2636 DELETE Q,PO
2638 DIM Q(2,3),PO(2)
2640 TP=6.283185*PA(1)*DT
2642 NS=50
2644 IF NS>N THEN NS=N
2646 DI=ITP(N/NS)
2650 I=IB
2652 AR=TP*I
2654 SN=SIN(AR)
2656 CO=COS(AR)
2658 Q(0,0)=Q(0,0)+SN*SN
2660 Q(0,1)=Q(0,1)+SN*CO
2662 Q(0,2)=Q(0,2)+SN
2664 Q(0,3)=Q(0,3)+Y(I)*SN
2666 Q(1,1)=Q(1,1)+CO*CO
2668 Q(1,2)=Q(1,2)+CO
2670 Q(1,3)=Q(1,3)+Y(I)*CO
2672 Q(2,2)=Q(2,2)+1
2674 Q(2,3)=Q(2,3)+Y(I)
2676 I=I+DI
2678 IF I<IE THEN 2652
2681 REM
2682 REM SOLVE SYSTEM FOR PARAMETER ESTIMATES
2683 REM
2684 P=3
2685 GOSUB 3636
2686 P=4
2687 PA(0)=SQR(Q(0,3)*Q(0,3)+Q(1,3)*Q(1,3))
2688 PA(3)=Q(2,3)
2689 PA(2)=SGN(Q(1,3))*1.570796
```

```
2690 IF Q(0,3)<>0 THEN PA(2)=ATN(Q(1,3)/Q(0,3))
2691 DELETE Q,PO
2692 RETURN
2693 REM **** END OF INITIAL PARAMETER ESTIMATION ROUTINE  ****
2694 REM
2700 REM **** ROUTINE TO FIND INITIAL PARAMETER ESTIMATES FOR ****
2702 REM **** NONLINEAR LEAST SQUARES SINE WAVE FIT TO DATA   ****
2704 REM **** CONTAINING AT LEAST ONE CYCLE.                 ****
2706 REM
2708 REM INPUTS:  IB,IE,N,DT,PA(1),Y
2710 REM OUTPUTS:  PA,CS
2712 REM
2714 REM NEW OR OVERWRITTEN VARIABLES: MS,PA(0),PA(2),PA(3),RS,CS,KI
2716 REM                               ZC,IX,ZI,T,KT
2718 REM
2720 REM GET MEAN,RAW AND CORRECTED SUM OF SQUARES, AMP. OF SINE WAVE
2722 REM BASED ON A (NEAR) INTEGER NUMBER OF CYCLES.
2724 TP=1/PA(1)
2726 KT=ITP(N*DT/TP)
2728 NS=ITP(KT*TP/DT)
2730 IF NS>N THEN NS=N
2732 IF NS=0 THEN NS=N
2734 IC=IB+NS-1
2736 REM BASE OFFSET, AMPLITUDE ESTIMATES ON POINTS IB THRU IC.
2738 PA(3)=MEA(Y(IB:IC))
2740 RS=RMS(Y(IB:IC))
2742 CS=NS*RS*RS-NS*PA(3)*PA(3)
2744 PA(0)=SQR(2*CS/NS)
2746 REM NOW FIND MEAN, S.S., ETC. OF ENTIRE DATA SEGMENT
2748 MS=MEA(Y(IB:IE))
2750 RS=RMS(Y(IB:IE))
2752 RS=N*RS*RS
2754 CS=RS-N*MS*MS
2756 REM REVISE FREQUENCY ESTIMATE
2758 GOSUB 2907
2760 REM
2762 REM ESTIMATE PHASE BY SIMPLE LINEAR SEARCH OVER [-PI,PI], FOLLOWED
2764 REM BY AN INTERPOLATION TO REFINE THE PHASE ESTIMATE. ONLY USE A
2766 REM SEGMENT OF THE WHOLE RECORD.
2768 NS=50
2770 IF NS<ITP(1/(DT*PA(1))) THEN NS=ITP(1/(DT*PA(1)))
2772 IF NS>N THEN NS=N
```

@

```
2774 REM SET INTERVAL TO SEARCH, SET RES. SUM OF SQ. TO RAW S.S.
2776 IR=3.141593\IL=-IR
2778 NT=16\SI=(IR-IL)/NT
2780 SM=RS
2782 REM
2784 GOSUB 2955
2786 REM USE PHASE OF LEAST RES. SUM OF SQUARES AND TWO ADJACENT
2788 REM PHASE POINTS TO REFINE PHASE ESTIMATE BY INTERPOLATION
2790 HC=SM
2792 IC=PA(2)
2794 IL=IC+SI
2796 NT=0
2798 SM=RS
2800 GOSUB 2955
2802 HR=SM
2804 IR=IL
2806 IL=IC-SI
2808 SM=RS
2810 GOSUB 2955
2812 HL=SM
2814 REM REFINE PHASE ESTIMATE BY FITTING A QUADRATIC AND FINDING
2816 REM ITS POINT OF MINIMUM.
2818 H1=(HR-HL)/(2*SI)
2820 H2=(HR-2*HC+HL)/(2*SI*SI)
2822 PA(2)=IC-H1/H2
2824 REM
2826 RETURN
2828 REM **** END ROUTINE TO FIND INITIAL PARAMETER ESTIMATES ****
2830 REM
2900 REM **** ROUTINE TO REVISE USER DEFINED FREQUENCY ESTIMATE ****
2901 REM INPUTS: PA(1),MS,Y,IB,IE,DT
2902 REM OUTPUTS: REVISED FREQ. ESTIMATE IN PA(1)
2903 REM OVERWRITTEN VARIABLES: TP,IX,KI,ZR,ZL,PA(1)
2904 REM
2905 REM FIND AVERAGE OF HALF-PERIODS FROM MEAN LEVEL CROSSINGS,
2906 REM DOUBLE IT, THEN FIND RECIPROCAL.
2907 TP=0
2908 IX=IB
2909 KI=0
2910 ZR=CRS(Y(IX:IE),MS)
2911 IF ZR=-1 THEN RETURN
2912 REM
```

```
2913 ZL=ZR
2914 IX=ITP(ZR)+1
2915 IF IX>IE THEN 2922
2916 ZR=CRS(Y(IX:IE),MS)
2917 IF ZR=-1 THEN 2922
2918 IF ZR-ZL<1/(1Ø*PA(1)*DT) THEN 2914
2919 KI=KI+1
292Ø TP=TP+(ZR-ZL)*DT
2921 GOTO 2913
2922 IF KI>Ø THEN PA(1)=1/(2*TP/KI)
2923 RETURN
2924 REM **** END FREQUENCY ESTIMATION ROUTINE  ****
2925 REM
295Ø REM **** LINEAR SEARCH ROUTINE TO GET INITIAL PHASE ESTIMATE ****
2951 REM INPUTS:IB,NS,IL,SI,NT,SM,Y,PA,DT
2952 REM OUTPUTS: "BEST" PHASE FOUND IN PA(2)
2953 REM NEW OR OVERWRITTEN VARIABLES: IG,BP,OM,I,SH,PA(2),TI,H1
2954 REM
2955 IG=IB+NS-1
2956 BP=IL
2957 OM=6.283185*PA(1)
2958 REM LINEAR SEARCH LOOP
2959 FOR I=Ø TO NT
296Ø SH=Ø
2961 PA(2)=IL+I*SI
2962 REM RESIDUAL SUM OF SQUARES LOOP
2963 FOR TI=IB TO IG
2964 H1=Y(TI)-(PA(Ø)*SIN(OM*TI*DT+PA(2))+PA(3))
2965 SH=SH+H1*H1
2966 NEXT TI
2967 REM
2968 IF SH>=SM THEN GOTO 2971
2969 SM=SH
297Ø BP=PA(2)
2971 NEXT I
2972 PA(2)=BP
2973 RETURN
2974 REM *** END ROUTINE TO LINEARLY SEARCH FOR INITIAL PHASE EST. ****
2975 REM
3ØØØ REM **** NONLINEAR LEAST SQUARES FIT ROUTINE  ****
3ØØ2 REM ****  USING NEWTON'S METHOD                ****
3ØØ4 REM
```

```
3006 REM ON ENTRY HAVE:
3008 REM P--NUMBER OF PARAMETERS IN MODEL OR FUNCTION
3010 REM PA--PARAMETER VECTOR CONTAINING INITIAL PARAMETER ESTIMATES
3012 REM Y--ARRAY CONTAINING DATA. DATA TO BE FIT IS ACTUALLY Y(IB),
3014 REM ---Y(IB+1),...,Y(IE).
3015 REM DT--DATA SAMPLING INTERVAL IF DATA IS EVENLY SPACED, ZERO
3016 REM ----OTHERWISE.
3017 REM IB,IE--BEGINNING AND END INDICES AS DEFINED ABOVE.
3018 REM CS--DATA SUM OF SQUARES, CORRECTED FOR THE MEAN
3020 REM IO--IF IO<>0 PARAMETER VALUES, RESIDUAL SUM OF SQUARES,
3021 REM ----AND R-SQUARED VALUES WILL BE PRINTED AT EACH ITERATION.
3022 REM NI--MAXIMUM NUMBER OF ITERATIONS TO TRY.
3024 REM PE--ARRAY OF LENGTH P CONTAINING THE CONVERGENCE LIMITS FOR
3025 REM ----EACH OF THE P PARAMETERS. CONVERGENCE OCCURS
3026 REM ----WHEN THE FOLLOWING CONDITIONS ARE SATISFIED:
3027 REM ----  ABS((PA(I)-PO(I))/PA(I))<PE(I) IF ABS(PA(I))>=PE(I)
3028 REM ---- OR ABS(PA(I)-PO(I))<PE(I) IF ABS(PA(I))<PE(I)
3029 REM ----WHERE PA AND PO ARE THE CURRENT AND PREVIOUS PARAMETER
3030 REM ----ESTIMATES AND I=0,1,...,P-1.
3031 REM ES--CONVERGENCE ALSO SAID TO HAVE OCCURRED IF:
3032 REM ----  ABS((SS-ST)/SS)<ES, WHERE SS AND ST ARE THE CURRENT AND
3033 REM ----PREVIOUS RESIDUAL SUM OF SQUARES RESPECTIVELY.
3034 REM NOTE: A USER ROUTINE DEFINING THE FUNCTION TO BE FIT, ALONG WITH
3035 REM       ITS PARTIAL DERIVATIVES WITH RESPECT TO EACH PARAMETER
3036 REM       MUST BE SUPPLIED. SEE LINES 3500-3599 FOR AN EXAMPLE. THE
3037 REM       ENTRY POINT OF THIS ROUTINE MUST BE LINE 3520.
3038 REM ON EXIT FIND:
3039 REM PA(0),PA(1),...,PA(P-1)--THE FINAL PARAMETER ESTIMATES.
3040 REM PE(0),PE(1),...,PE(P-1)--ESTIMATED STANDARD DEVIATION OF THE P
3041 REM ----PARAMETER ESTIMATES.
3042 REM CM--THE PARAMETER COVARIANCE MATRIX (PXP).
3045 REM SS--FINAL RESIDUAL SUM OF SQUARES.
3046 REM S-- STANDARD DEVIATION OF THE RESIDUALS.
3047 REM EC--ERROR CODE. EC=0 IF ONE OF THE CONVERGENCE CRITIERIA ABOVE
3048 REM ----WAS SATISFIED WITHIN THE NI ITERATIONS ALLOWED, OR EC=1 IF
3049 REM ----NO CONVERGENCE IN NI ITERATIONS.
3051 REM
3052 REM VARIABLE NAMES USED: P,PA,Y,IB,IE,CS,IO,NI,EP,PE,SS,S,EC,Q,CM
3053 REM ---------  PO,N,IC,CP,I,HP,PE
3054 REM
3056 REM
3062 REM
```

```
3063 REM
3064 REM SET UP MATRIX Q, THE AUGMENTED COEFFICIENT MATRIX OF THE
3065 REM LINEARIZED SYSTEM; THE COVARIANCE MATRIX; AND A SCRATCH
3066 REM ARRAY, PO; FINALLY, AN ARRAY FOR PARTIAL DERIVATIVES, HP.
3067 DELETE Q,CM,PO,HP
3068 DIM Q(P-1,P),CM(P-1,P-1),PO(P-1),HP(P-1)
3069 N=IE-IB+1
3072 ST=CS
3073 IC=0
3075 IF IO<>0 THEN GOSUB 3260
3076 REM BUILD Q MATRIX, GET RESIDUAL S.S. FOR INITIAL PARA. ESTIMATES
3077 GOSUB 3420
3078 IF IO<>0 THEN GOSUB 3280
3079 REM
3080 REM ---------  NEWTON ITERATION  ---------
3081 REM
3082 REM SOLVE THE SYSTEM OF EQUATIONS
3084 GOSUB 3636
3086 REM GET NEW PARAMETER ESTIMATES
3088 PO=PA
3089 PA=PO+Q(0:P-1,P)
3090 REM BUILD NEW Q MATRIX, GET NEW RESIDUAL S.S.
3091 GOSUB 3420
3092 IC=IC+1
3093 IF IO<>0 THEN GOSUB 3280
3094 REM TEST FOR CONVERGENCE
3099 GOSUB 3320
3100 ST=SS
3101 IF EC=0 THEN GOTO 3110
3102 IF IC<NI THEN GOTO 3084
3103 REM
3104 REM --------- END OF NEWTON ITERATONS  ------------
3106 REM
3108 REM CONSTRUCT PARAMETER COVARIANCE  MATRIX
3110 GOSUB 3734
3112 S=SS/(N-P)
3114 CM=S*CM
3116 S=SQR(S)
3120 FOR I=0 TO P-1
```

```
3122 PE(I)=SQR(CM(I,I))
3124 NEXT I
3126 RETURN
3128 REM  **** END NONLINEAR LEAST SQUARES FIT ROUTINE  ****
3129 REM
3250 REM ****  INITIAL PRINT-OUT ROUTINE  ****
3252 REM INPUTS: P,CS
3254 REM OUTPUTS: PRINT-OUT
3258 REM
3260 PRINT
3262 PRINT "NO. OF PARAMETERS=";P
3264 PRINT "DATA SUM OF SQUARES (CORRECTED)=";CS
3266 RETURN
3268 REM ****  END INITIAL PRINT-OUT ROUTINE  ****
3269 REM
3270 REM ****  ROUTINE TO PRINT OUT INTERMEDIATE RESULTS  ****
3271 REM
3272 REM INPUTS: IC,P,PA,SS,CS
3274 REM OUTPUTS: PRINT-OUT AND R2 VALUE
3275 REM NEW OR OVERWRITTEN VARIABLES: THE R-SQUARED VALUE R2
3278 REM
3280 R2=1-SS/CS
3282 PRINT
3284 PRINT "ITERATION=";IC,"RESIDUAL S.S.=";SS,"R^2 VALUE=";R2
3286 PRINT "PARAMETER VALUES:";PA
3288 RETURN
3290 REM **** END INTERMEDIATE PRINT-OUT ROUTINE  ****
3292 REM
3300 REM **** ROUTINE TO TEST FOR CONVERGENCE   ****
3301 REM
3302 REM INPUTS: P,PA,PO,PE,SS,ST,ES
3304 REM OUTPUTS: ERROR CODE EC
3305 REM NEW OR OVERWRITTEN VARIABLES: EC,I,ER
3306 REM
3320 EC=1
3322 REM TEST FOR CHANGE IN RESIDUAL SUM OF SQUARES
3324 IF ABS((SS-ST)/SS)>=ES THEN GOTO 3352
3326 EC=0
3328 RETURN
3330 REM
3346 REM CHECK EACH PARAMETER FOR CONVERGENCE CRITERION
3352 FOR I=0 TO P-1
```

```
3354 ER=ABS(PA(I)-PO(I))
3356 IF ABS(PA(I))>=PE(I) THEN ER=ER/ABS(PA(I))
3358 IF ER>=PE(I) THEN RETURN
3360 NEXT I
3362 EC=0
3364 RETURN
3366 REM **** END CONVERGENCE TEST ROUTINE  ****
3368 REM
3400 REM **** BUILD Q-- THE AUGMENTED COEFFICIENT MATRIX  ****
3402 REM INPUTS: P,IB,N
3404 REM OUTPUTS: Q AND SS
3406 REM NEW OR OVERWRITTEN VARIABLES: SS,Q,TI,ER,I,J
3408 REM
3420 Q=0
3422 SS=0
3424 REM
3425 FOR TI=IB TO IE
3428 GOSUB 3530
3429 ER=Y(TI)-H
3430 SS=SS+ER*ER
3432 FOR I=0 TO P-1
3434 Q(I,P)=Q(I,P)+ER*HP(I)
3436 FOR J=I TO P-1
3438 Q(I,J)=Q(I,J)+HP(I)*HP(J)
3440 NEXT J
3442 NEXT I
3444 NEXT TI
3446 RETURN
3448 REM  **** END ROUTINE TO BUILD Q MATRIX  ****
3449 REM
3500 REM **** EVALUATES USER FUNCTION, ITS PARTIAL DERIVATIVES ****
3501 REM ****  GIVEN A CURRENT TIME AND PARAMETER VECTOR       ****
3502 REM
3503 REM **** SAMPLE USER ROUTINE: SINE FUNCTION  ****
3504 REM ON ENTRY HAVE:
3506 REM P--NUMBER OF PARAMETERS
3508 REM PA--PARAMETER VECTOR CONTAINING CURRENT
3509 REM ----ESTIMATES OF THE P PARAMETERS.
3510 REM TI--INDEX OF THE CURRENT TIME (IB THRU IE)
3512 REM DT--THE DATA SAMPLING INTERVAL
3513 REM HP--AN ARRAY OF LENGTH P FOR PARTIAL DERIVATIVES.
3514 REM ON EXIT HAVE:
```

```
3516 REM H--THE VALUE OF THE FUNCTION FOR THE CURRENT TIME AND SET OF
3518 REM ---PARAMETER VALUES.
352Ø REM HP(Ø),HP(1),...,HP(P-1)--THE PARTIAL DERIVATIVES WITH
3522 REM ----RESPECT TO EACH PARAMTETER, EVALUATED AT THE CURRENT
3523 REM ----TIME AND PARAMETER VALUES.
3524 REM
3526 REM NEW OR OVERWRITTEN VARIABLES: T,H,HP,H1,AR,P2
3528 REM
353Ø P2=6.283185
3531 T=TI*DT
3532 AR=P2*PA(1)*T+PA(2)
3534 H1=SIN(AR)
3536 H=PA(Ø)*H1+PA(3)
3538 REM
354Ø HP(Ø)=H1
3542 H1=COS(AR)
3544 HP(1)=PA(Ø)*P2*T*H1
3546 HP(2)=PA(Ø)*H1
3548 HP(3)=1
355Ø RETURN
3552 REM ****  END USER FUNCTION ROUTINE   ****
3555 REM
36ØØ REM **** ROUTINE TO SOLVE SYSTEM OF EQUATIONS WITH POSITIVE    ****
36Ø2 REM **** DEFINITE AND SYMMETRIC COEFFICIENT MATRIX USING       ****
36Ø4 REM **** CHOLESKY'S METHOD. SUITABLE FOR SYSTEMS OF LOW ORDER. ****
36Ø6 REM **** INNER PRODUCT SUMMING IS DONE DOUBLE PRECISION        ****
36Ø8 REM
361Ø REM ON ENTRY HAVE:
3612 REM P--THE MATRIX TO INVERT IS PXP, AUGMENTED IT IS PX(P+1).P>=2.
3614 REM Q--THE AUGMENTED COEFFICIENT MATRIX; Q=(R,Y) WHERE R
3616 REM ---IS THE COEFFICIENT MATRIX. THE DECOMPOSITION IS
3618 REM ---R=TRANS(U)*U, U UPPER TRIANGULAR. ONLY THE UPPER TRIANGULAR
362Ø REM ---PORTION OF Q IS USED.
3622 REM PO--A SCRATCH ARRAY OF LENGTH AT LEAST P-1.
3624 REM ON EXIT FIND:
3626 REM Q--CONTAINING THE PX(P+1) MATRIX WITH ELEMENTS (U,Z), WHERE Z
3628 REM ---IS THE SOLUTION VECTOR. ONLY UPPER TRIANGULAR ELEMENTS VALID.
363Ø REM NEW OR OVERWRITTEN VARIABLES: Q,KL,I,J,ER,K,QD,PO
3632 REM THE CHOLESKY DECOMPOSITION
3634 REM
3636 IF P>=2 THEN GOTO 3644
3638 PRINT "CHOLESKY SYSTEM SOLVER ERROR: P TOO SMALL."
```

```
3640 STOP
3642 REM THE CHOLESKY DECOMPOSITION ON AN AUGMENTED MATRIX
3644 IF Q(0,0)=0 THEN GOSUB 3854
3646 Q(0,0)=SQR(Q(0,0))
3648 QD=1/Q(0,0)
3650 Q(0,1:P)=QD*Q(0,1:P)
3652 REM
3654 FOR I=1 TO P-1
3656 FOR J=I TO P
3658 PO(0:I-1)=Q(0:I-1,I)*Q(0:I-1,J)
3660 ER=I*MEA(PO(0:I-1))
3662 Q(I,J)=Q(I,J)-ER
3664 NEXT J
3666 IF Q(I,I)<=0 THEN GOSUB 3854
3668 Q(I,I)=SQR(Q(I,I))
3670 QD=1/Q(I,I)
3672 Q(I,I+1:P)=Q(I,I+1:P)*QD
3674 NEXT I
3676 REM
3678 REM NOW BACK SUBSTITUTE:
3680 GOSUB 3812
3682 RETURN
3684 REM ****  END ROUTINE TO SOLVE SYSTEM OF EQUATIONS   ****
3686 REM
3700 REM **** ROUTINE TO INVERT A SYMMETRIC, POSITIVE DEFINITE   ****
3702 REM **** MATRIX USING CHOLESKY DECOMPOSITION APPROACH       ****
3704 REM **** SUITABLE FOR SMALL MATRICES. INNER PRODUCT SUMMING ****
3706 REM **** DONE DOUBLE PRECISION.                             ****
3708 REM
3710 REM ON ENTRY HAVE:
3712 REM P--SET TO THE DIMENSION OF THE MATRIX TO INVERT.
3714 REM Q--A PX(P+1) MATRIX WITH MATRIX TO INVERT IN FIRST P COLUMNS.
3716 REM ---ONLY THE UPPER TRIANGULAR PORTION IS USED IN THE COMPUTATIONS
3718 REM CM--A PXP MATRIX TO STORE INVERSE.
3720 REM ON EXIT FIND:
3722 REM CM--CONTAINING THE INVERSE MATRIX.
3724 REM NEW OR OVERWRITTEN VARIABLES: Q,CM,I,J,ER
3726 REM
3728 REM INVERT BY SOLVING P SYSTEMS OF EQUATION WITH AUGMENTED MATRIX
3730 REM Q=(R,Y), WHERE Y IS EACH OF THE P COLUMN OF THE IDENTITY MATRIX
3732 REM IN TURN. START WITH Y EQUAL TO FIRST COLUMN.
3734 IF P>=2 THEN GOTO 3740
```

@

```
3736 PRINT "INVERT ROUTINE ERROR: P TOO SMALL."
3738 STOP
3740 Q(0:P-1,P)=0
3742 Q(0,P)=1
3744 GOSUB 3636
3746 CM(0:P-1,0)=Q(0:P-1,P)
3748 REM NOW SOLVE THE LAST SYSTEM
3750 Q(0:P-2,P)=0
3752 Q(P-1,P)=1/Q(P-1,P-1)
3754 GOSUB 3812
3756 CM(0:P-1,P-1)=Q(0:P-1,P)
3758 REM NOW SOLVE THE REMAINING P-2 SYSTEMS
3760 IF P=2 THEN RETURN
3762 FOR J=1 TO P-2
3764 Q(0:J-1,P)=0
3766 Q(J,P)=1/Q(J,J)
3768 FOR I=J+1 TO P-1
3770 PO(J:I-1)=Q(J:I-1,P)*Q(J:I-1,I)
3772 ER=(I-J)*MEA(PO(J:I-1))
3774 Q(I,P)=-ER/Q(I,I)
3776 NEXT I
3778 REM BACK SUBSTITUTE TO FINISH SOLVING J-TH SYSTEM
3780 GOSUB 3812
3782 REM STORE THE J-TH SYSTEM SOLUTION
3784 CM(0:P-1,J)=Q(0:P-1,P)
3786 NEXT J
3788 RETURN
3790 REM ****  END MATRIX INVERT ROUTINE   ****
3792 REM
3800 REM ****  BACK SUBSTITUTION ROUTINE FOR AUGMENTED Q MATRIX  ****
3802 REM INPUTS: Q MATRIX(PX(P+1)) IN UPPER TRIANGULAR FORM; SCRATCH
3804 REM         ARRAY PO OF LENGTH AT LEAST P.
3806 REM OUTPUTS: SOLUTION VECTOR STORED IN (P+1)-ST COLUMN OF Q.
3808 REM NEW OR OVERWRITTEN VARIABLES: I,ER, AND (P+1)-ST COLUMN OF Q
3810 REM
3812 Q(P-1,P)=Q(P-1,P)/Q(P-1,P-1)
3814 FOR I=P-2 TO 0 STEP -1
3816 PO(I+1:P-1)=Q(I,I+1:P-1)*Q(I+1:P-1,P)
3818 ER=(P-I-1)*MEA(PO(I+1:P-1))
3820 Q(I,P)=(Q(I,P)-ER)/Q(I,I)
3822 NEXT I
3824 RETURN
```

```
3826 REM **** END BACK SUBSTITUTION ROUTINE  ****
3828 REM
3850 REM **** ERROR ROUTINE ---- SINGULAR Q MATRIX ****
3852 REM
3854 PRINT "SINGULAR Q-MATRIX ON ITERATION ";IC;" ."
3856 STOP
3858 REM **** END ERROR ROUTINE ****
3860 REM
4000 REM **** ROUTINE TO PRINT RESULTS OF SINE WAVE FIT  ****
4005 REM
4010 REM INPUTS: PA,PE,R2,EC
4015 REM
4020 PRINT
4025 PRINT "FINAL PARAMETER ESTIMATES:"
4030 PRINT "AMP=";PA(0);"  FREQ=";PA(1);"  PHASE=";PA(2);"  DC=";PA(3)
4035 PRINT "PARAMETER STANDARD ERRORS:"
4040 PRINT "    ";PE(0);"       ";PE(1);"        ";PE(2);"    ";PE(3)
4045 PRINT
4050 PRINT "R^2 VALUE=";R2
4055 PRINT "ERROR CODE=";EC
4060 REM
4062 RETURN
4065 REM ****  END FINAL PARAMETER PRINT-OUT  ****
4070 REM
4075 REM
```

## (MDPSS.BAS) PROGRAM 5-40. DYNAMIC PERFORMANCE TESTING OF DIGITIZERS

```
2500 REM **** MDPSS.BAS    ROUTINE TO FOLLOW MDPRMP.BAS OR  ****
2502 REM **** MDPSIN.BAS.  COMPUTES SUMMARY STATISTICS,     ****
2504 REM **** POSSIBLY WRITES DATA FILES FOR LATER ANALYSIS ****
2506 REM
2508 REM INPUTS:ST$,ID$,FN$,N,CS,IB,IE,NB,Y,DT,HU$,VU$,MS,AH,BH,PA
2510 REM OUTPUTS: PRINT-OUT OF SUMMARY STATISTICS AND DATA FILES
2512 REM OVERWRITTEN VARIABLES:RY,WR,SY,WS,MX,MN,RR,RQ,EB,RF,SN,RM
2514 REM
2516 REM ASK FOR OUTPUT FILE NAME
2550 PRINT "^G^G"
2552 PRINT "ENTER UP TO FOUR CHARACTERS FOR A FILE NAME FOR OUTPUT OF";
2554 PRINT " DATA FOR LATER"
2556 PRINT "ANALYSIS. ENTER THE STRING 'NULL' FOR NO FILE OUTPUT"
2558 INPUT FO$
2562 REM
2564 REM
2566 REM SET UP NECESSARY ARRAYS AND FILE NAMES
2568 GOSUB 3014
2570 DELETE WR,RY
2572 WAVEFORM WR IS RY(N-1),DT,HU$,VU$
2574 RY=Y(IB:IE)
2576 REM
2578 REM BEGIN PRINTING SUMMARY STATISTICS, WRITIING OUT DATA
2580 PAGE
2582 PRINT "I.D.  ";ID$
2584 PRINT "SIGNAL TYPE:  ";ST$
2586 PRINT "FILE NAME:  ";FN$
2588 PRINT
2590 PRINT "******** SUMMARY STATISTICS (IN ";VU$;" OR AS INDICATED) ";
2592 PRINT "********"
2594 REM DO "DC" TYPE SIGNALS SEPARATELY
2596 IF ST$="DC" THEN 2734
2598 PRINT
2600 PRINT "RMS OF DIGITIZER OUTPUT(DC REMOVED) = ";SQR(CS/N)
2602 REM NOW WRITE OUT DIGITIZER DATA, ITS LABEL
2604 IF FO$="NULL" THEN 2614
2606 WRITE #2,"DIGITIZER OUTPUT"
2608 WRITE #2,WR
2610 REM
2612 REM FIND (SYNTHESIZED) ANALOG INPUT, ANALOG ERRORS
2614 IF ST$="RAMP" THEN GOSUB 3220
2616 IF ST$="SINE" THEN GOSUB 3420
```

```
2618 REM
262Ø REM WRITE OUT ANALOG INPUT, ANALOG ERRORS
2622 IF FO$="NULL" THEN 2628
2624 WRITE #4,"ANALOG INPUT (SYNTHESIZED)"
2626 WRITE #4,WR
2628 RM=MEA(RY)
263Ø RY=RY-RM
2632 RF=RMS(RY)
2634 RY=Y(IB:IE)-RY-RM
2636 IF FO$="NULL" THEN 2646
2638 WRITE #3,"ANALOG ERRORS FOR DIGITIZER"
264Ø WRITE #3,WR
2642 REM
2644 REM PRINT ANALOG ERRORS, STATS
2646 PRINT
2648 MX=MAX(RY)
265Ø MN=MIN(RY)
2652 RR=RMS(RY)
2654 PRINT "ANALOG ERRORS FOR DIGITIZER OUTPUT:"
2656 PRINT "      MAX = ";MX
2658 PRINT "      MIN = ";MN
266Ø PRINT "      RMS = ";RR
2662 REM
2664 REM COMPUTE THE SIGNAL-TO-NOISE RATIO
2666 PRINT
2668 SN=2Ø*LOG(RF/RR)/LOG(1Ø)
267Ø PRINT "SIGNAL-TO-NOISE RATIO = ";SN;" DB"
2672 REM
2674 REM RE-SYNTHESIZE ANALOG INPUT
2676 IF ST$="RAMP" THEN GOSUB 322Ø
2678 IF ST$="SINE" THEN GOSUB 342Ø
268Ø REM GET OUTPUT OF IDEAL DIGITIZER, ITS QUANTIZATION ERROR
2682 GOSUB 3618
2684 REM
2686 REM ANALOG ERRORS FOR IDEAL DIG.(QUANT. ERRORS)
2688 IF FO$="NULL" THEN 2696
269Ø WRITE #5,"ERRORS OF IDEAL DIGITIZER"
2692 WRITE #5,WR
2694 REM
2696 RQ=RMS(RY)
2698 PRINT
27ØØ PRINT "RMS ERROR FOR IDEAL DIGITIZER = ";RQ
```

```
2702 REM
2704 REM GET EFFECTIVE BITS FIGURE
2706 EB=NB-LOG(RR/RQ)/LOG(2)
2708 PRINT
2710 PRINT "EFFECTIVE BITS = ";EB
2712 RY=Y(IB:IE)
2714 REM
2716 REM WRITE OUT IDEAL DIGITIZER OUTPUT
2718 IF FO$="NULL" THEN 2778
2720 REM
2722 WRITE #6,"IDEAL DIGITIZER OUTPUT"
2724 WRITE #6,WR
2726 GOTO 2770
2728 REM ******* END BLOCK FOR RAMP, SINE   *******
2730 REM
2732 REM FOR "DC" TYPE SIGNALS
2734 IF FO$="NULL" THEN 2740
2736 WRITE #2,"DIGITIZER OUTPUT"
2738 WRITE #2,WR
2740 RY=RY-MS
2742 MX=MAX(RY)
2744 MN=MIN(RY)
2746 RR=RMS(RY)
2748 PRINT
2750 PRINT "ERRORS FOR DIGITIZER OUTPUT:"
2752 PRINT "      MAX = ";MX
2754 PRINT "      MIN = ";MN
2756 PRINT "      RMS = ";RR
2758 REM
2760 IF FO$="NULL" THEN 2778
2762 WRITE #3,"DIGITIZER ERRORS"
2764 WRITE #3,WR
2766 REM
2768 REM CLOSE FILES AND RETURN
2770 FOR I=1 TO FF+1
2772 CLOSE #I
2774 NEXT I
2776 REM
2778 DELETE WR,RY
2780 RETURN
2782 REM *******  END ROUTINE TO PRINT SUMMARY STATS, WRITE FILES  ******
2784 REM
```

```
3000 REM ******  ROUTINE TO SET UP FILE NAMES  ******
3002 REM
3004 REM INPUTS: FO$,DT,HU$,VU$,N,ID$,ST$,FN$,NB,F,ND,IB,IE,Y
3006 REM OUTPUTS: FILE NAMES FD$(0),...,FD$(FF) AND WAVEFORM WR
3008 REM OVERWRITTEN VARIABLES:FF,FD$
3010 REM
3012 REM DEFINE THE FILE NAMES, OPEN THE FILES
3014 IF FO$="NULL" THEN 3046
3016 FF=5
3018 IF ST$="DC" THEN FF=2
3020 DELETE FD$
3022 DIM FD$(FF)
3024 FOR I=0 TO FF
3026 FD$(I)=FO$&STR(I)&".DAT"
3028 CANCEL FD$(I)
3030 CLOSE #I+1
3032 OPEN #I+1 AS FD$(I) FOR WRITE
3034 NEXT I
3036 REM WRITE OUT HEADER FILE
3038 WRITE #1,ID$,ST$,FN$
3040 WRITE #1,NB,DT,F,ND,IB,IE
3042 IF ST$='SINE' THEN 3048
3044 WRITE #1,AH,BH
3046 GOTO 3050
3048 WRITE #1,PA
3050 RETURN
3052 REM ***** END FILE SET-UP ROUTINE  *******
3054 REM
3200 REM ****** ROUTINE TO GENERATE ANALOG INPUT SIGNAL: RAMP   ******
3204 REM
3206 REM INPUTS:IB,IE,BH,AH,DT
3208 REM OUTPUTS: RY, THE SYNTHESIZED ANALOG INPUT
3210 REM OVERWRITTEN VARIABLES: II,I,RY
3214 REM
3220 II=0
3225 FOR I=IB TO IE
3230 RY(II)=BH*DT*I+AH
3235 II=II+1
3240 NEXT I
3245 REM
3250 RETURN
3255 REM *********   END RAMP ROUTINE   ********
```



@

```
3260 REM
3400 REM **** ROUTINE TO GENERATE ANALOG INPUT SIGNAL OR ****
3402 REM **** DATA.  THIS ROUTINE FOR A SINE WAVE MODEL  ****
3404 REM
3406 REM INPUTS:IB,N,DT,PA, ARRAY RY OF LENGTH N.
3408 REM OUTPUTS:FITTED DATA IS ARRAY RY
3410 REM OVERWRITTEN VARIABLES: OM,I,RY
3412 REM
3420 OM=6.283185*PA(1)*DT
3425 FOR I=0 TO N-1
3430 RY(I)=PA(0)*SIN(OM*(I+IB)+PA(2))+PA(3)
3435 NEXT I
3440 RETURN
3445 REM ***** END SINE MODEL ROUTINE   *********
3450 REM
3600 REM **** ROUTINE TO EXACTLY DIGITIZE FROM FITTED MODEL DATA ****
3602 REM *** AND FIND QUANTIZATION ERRORS   *****
3604 REM INPUTS: NB,IB,IE,ARRAY Y OF LENGTH >=N=IE-IB+1,
3606 REM         FITTED DATA IN ARRAY RY OF LENGTH N.
3608 REM OUTPUT: IDEAL DIGITIZER OUTPUT IN Y(IB:IE)
3610 REM ERRORS IN RY.
3612 REM OVERWRITTEN VARIABLES: VU,VL,HV,HL,I,H,RY,Y
3614 REM
3616 REM SET UP FULL SCALE, OUT OF RANGE CODES
3618 VU=2^NB-.5\HV=2^NB-1
3620 VL=-.5\LV=0
3622 REM
3624 FOR I=0 TO N-1
3626 H=RY(I)
3628 Y(IB+I)=ITP(H+.5)
3630 IF H>=VU THEN Y(IB+I)=HV
3632 IF H<=VL THEN Y(IB+I)=LV
3634 RY(I)=Y(IB+I)-H
3636 NEXT I
3638 RETURN
3640 REM **** END ROUTINE TO EXACTLY DIGITIZE, FIND QUANTIZATION ERROR **
3642 REM
```

(MDPANL.BAS) PROGRAM 5-40. DYNAMIC PERFORMANCE TESTING OF DIGITIZERS

```
100 REM ****  MDPANL.BAS,  MAIN PROGRAM FOR OVERLAYING ROUTINES ****
105 REM ****  FOR GRAPHING OR ANALYZING DIGITIZER PERFORMANCE   ****
110 REM ****     TEK SPS BASIC, AUGUST 10, 1977                 ****
115 REM
120 REM ASK FOR DATA FILE NAME (FOUR CHARACTER MDP NAME )
125 PRINT
130 PRINT "ENTER THE FOUR CHARACTER 'MDP' DATA FILE NAME"
135 INPUT FO$
140 REM
145 REM SET UP CODE FOR OVERLAY CURRENTLY IN MEMORY
150 CO=0
155 REM
160 REM PRINT MENU, ASK FOR OPTION CODE
165 PAGE
170 COPY "MENU.MDP" TO KB:
175 PRINT
180 PRINT "ENTER OPTION CODE."
185 PRINT "^G^G"
190 INPUT CN
195 IF CN=-1 THEN 260
200 REM
205 REM OVERLAY ROUTINE IF NECESSARY, UPDATE "OLD" OVERLAY CODE
210 IF CN=CO THEN 240
215 DELETE 2500,4999
220 RELEASE ALL
225 FP$="MDPA"&STR(CN)&".BNR"
230 OVERLAY FP$
235 CO=CN
240 GOSUB 2550
245 REM
250 REM RECYCLE
255 GOTO 165
260 RETURN
265 REM
270 REM **** END PROGRAM MDPANL.BAS  ****
275 REM
280 REM
```

# MENU

OPTIONS

1--GRAPH DATA (1 OR 2 SETS)
2--COMPUTE AMP. SPECTRA (1 OR 2 SETS)
3--COMPUTE RMS TIME JITTER AND RMS
   RESIDUAL ADDITIVE ERROR
4--GRAPH RMS ANALOG ERROR VERSUS
   DIGITAL CODE.
5--GRAPH RMS ANALOG ERROR VERSUS PHASE

DATA

1--DIGITIZER OUTPUT
2--ANALOG ERRORS FOR DIGITIZER
3--ANALOG INPUT (SYNTHESIZED)
4--ERRORS FOR IDEAL DIGITIZER
5--IDEAL DIGITIZER OUTPUT

AFTER PROMPTING MESSAGES, ENTER OPTION CODE AND, IF NECESSARY, DATA CODES. AN OPTION CODE OF -1 TERMINATES THE ANALYSIS OPTIONS. A DATA CODE OF -1 TERMINATES ENTRY OF DATA CODES AT LESS THAN THE MAXIMUM NUMBER.

OPTIONS 3 AND 5 APPLY ONLY TO SINUSOIDAL SIGNALS.

OPTIONS 3,4, AND 5 DO NOT REQUIRE DATA CODES.

FOR DC-TYPE SIGNALS, ONLY DATA CODES 1 AND 2 ARE VALID.

```
25ØØ REM **********   MDPA1.BAS     ********************
25Ø5 REM **** ROUTINE TO INPUT AND GRAPH UP TO TWO DATA SETS ******
251Ø REM
2515 REM INPUTS: FOUR CHARACTER DATA FILE NAME FO$
252Ø REM OUTPUTS: GRAPHS ONLY
2525 REM OVERWRITTEN VARIABLES: FM,FF,FC$,WS,CC$,I1
253Ø REM
254Ø REM GET DATA CODES
255Ø FM=5
2552 FF=2
2554 DELETE FC$
2556 DIM FC$(FF-1)
2558 GOSUB 3Ø5Ø
256Ø IF NC=Ø THEN 2636
2562 REM
2564 REM READ IN HEADER FILE, SET UP SCRATCH WAVEFORM
2566 GOSUB 325Ø
2568 REM
257Ø REM READ IN FIRST DATA SET
2572 CC$=FC$(Ø)
2574 GOSUB 345Ø
2576 REM
2578 REM ONE OR TWO DATA SETS? HANDLE SEPARATELY
258Ø IF NC=2 THEN 2596
2582 PAGE
2584 PRINT "               ";LB$
2586 GRAPH WR
2588 GOTO 2624
259Ø REM
2592 REM TWO DATA SETS TO PLOT. SET UP, READ IN
2594 REM ANOTHER SET.
2596 DELETE WS,SY
2598 WAVEFORM WS IS SY(N-1),DT,HU$,VU$
26ØØ WS=WR
26Ø2 PL$=LB$
26Ø4 CC$=FC$(1)
26Ø6 GOSUB 345Ø
26Ø8 REM
261Ø PAGE
2612 PRINT "              ";PL$;"   ^^H_-";LB$
2614 SETGR NOPLOT
2616 GRAPH WS,WR
```

```
2618 DISPLAY WS
2620 DISPLAY 3,FM,WR
2622 REM
2624 PRINT "^G^G"
2626 WAIT
2628 REM
2630 REM CLEAN HOUSE A LITTLE
2632 DELETE RY,SY,WR,WS
2634 REM
2636 RETURN
2638 REM ****** END PLOT ROUTINE  ********
2640 REM
2642 REM
3000 REM *****  ROUTINE TO INPUT DATA CODES  ******
3005 REM
3010 REM INPUTS:FF AND ARRAY FC$
3015 REM OUTPUTS: DATA CODES IN FC$(0),...,FC$(FF-1), AND NC
3020 REM OVERWRITTEN VARIABLES: I1$,NC
3025 REM
3050 NC=0
3052 PRINT
3054 PRINT "ENTER DATA CODE(S) ONE PER LINE, TERMINATE WITH -1 CODE"
3055 INPUT I1$
3060 IF I1$="-1" THEN 3090
3065 FC$(NC)=I1$
3070 NC=NC+1
3075 IF NC>=FF THEN 3090
3080 GOTO 3055
3085 REM
3090 RETURN
3095 REM  **** END ROUTINE TO INPUT DATA CODES   *****
3100 REM
3105 REM
3200 REM **** ROUTINE TO READ HEADER FILE, SET UP SCRATCH WAVEFORM ****
3205 REM
3210 REM INPUTS:FO$ (FOUR CHARACTER 'MDP' DATA FILE NAME)
3215 REM OUTPUTS: ID$,ST$,FN$,NB,DT,F,ND,N,HU$,VU$
3220 REM OVERWRITTEN VARIABLES: I1$ AND OUTPUTS
3225 REM
3250 CLOSE #1
3255 I1$=FO$&"0"&".DAT"
3260 OPEN #1 AS I1$ FOR READ
```

```
3265 READ #1,ID$,ST$,FN$
3270 READ #1,NB,DT,F,ND,IB,IE
3275 CLOSE #1
3280 N=IE-IB+1
3285 REM
3290 DELETE RY,WR
3295 HU$="S"
3300 VU$="LSB"
3305 WAVEFORM WR IS RY(N-1),DT,HU$,VU$
3310 REM
3315 RETURN
3320 REM ****** END ROUTINE TO READ HEADER FILE, SET UP WAVEFORM  *****
3325 REM
3330 REM
3400 REM *****  ROUTINE TO READ IN A DATA SET   ******
3405 REM
3410 REM INPUTS:FO$; CURRENT DATA CODE IN CC$; SCRATCH WAVEFORM WR
3415 REM OUTPUTS:LB$, LABEL FOR DATA; DATA IN WAVEFORM WR.
3420 REM OVERWRITTEN VARIABLES WR,LB$
3425 REM
3450 I1$=FO$&CC$&".DAT"
3455 OPEN #1 AS I1$ FOR READ
3460 READ #1,LB$
3465 READ #1,WR
3470 CLOSE #1
3475 REM
3480 RETURN
3485 REM ***** END ROUTINE TO READ IN A DATA SET  ****
3490 REM
```

(MDPA2.BAS) PROGRAM 5-40. DYNAMIC PERFORMANCE TESTING OF DIGITIZERS

```
2500 REM **** MDPA2.BAS ROUTINE TO COMPUTE AND GRAPH AMPLITUDE   ****
2502 REM **** SPECTRA OF ONE OR TWO DATA SETS PREVIOUSLY WRITTEN ****
2504 REM **** ON DISK BY THE MDPSS.BAS PROGRAM.                  ****
2506 REM
2508 REM INPUTS: FOUR CHARACTER DATA FILE NAME FO$
2510 REM OUTPUTS: GRAPHS ONLY
2512 REM OVERWRITTEN VARIABLES: FM,FF,FC$,CC$,W1,W2,A1,A2
2514 REM
2516 REM GET DATA CODES
2550 FM=5
2552 FF=2
2554 DELETE FC$,W1,A1,W2,A2
2556 DIM FC$(FF-1)
2558 GOSUB 3012
2560 IF NC=0 THEN 2642
2562 REM
2564 REM READ IN HEADER FILE, SET UP SCRATCH WAVEFORM
2566 GOSUB 3112
2568 REM
2570 REM READ IN FIRST DATA SET
2572 CC$=FC$(0)
2574 GOSUB 3212
2576 REM
2578 REM GET AMPLITUDE SPECTRUM
2586 WAVEFORM W1 IS A1(N/2),D1,H1$,V1$
2588 GOSUB 3318
2590 REM READ IN AND TRANSFORM A SECOND SET OF DATA?
2592 IF NC=2 THEN 2608
2594 PAGE
2596 PRINT "AMPLITUDE SPECTRUM:   ";LB$
2597 PRINT "(MEAN REMOVED, THEN ";CW;" % COSINE WINDOWED)"
2598 GRAPH W1
2600 GOTO 2636
2602 REM
2604 REM READ IN AND TRANSFORM A SECOND SET OF DATA
2608 WAVEFORM W2 IS A2(N/2),D2,H2$,V2$
2610 W2=W1
2611 PL$=LB$
2612 CC$=FC$(1)
2614 GOSUB 3212
2616 GOSUB 3318
2618 DELETE WR,RY
```

```
262Ø PAGE
2622 PRINT "AMPLITUDE SPECTRA (";CW;" % COSINE TAPER USED):"
2624 PRINT "       ";PL$;"   ^^H_-";LB$
2626 SETGR NOPLOT
2628 GRAPH W2,W1
263Ø DISPLAY W2
2632 DISPLAY 3,FM,W1
2634 REM
2636 PRINT "^G^G"
2638 WAIT
264Ø DELETE W1,A1,W2,A2
2642 RETURN
2644 REM **** END OF ROUTINE MDPA2.BAS ****
2646 REM
3ØØØ REM **** ROUTINE TO INPUT DATA CODES, % COSINE TAPER ****
3ØØ2 REM
3ØØ4 REM INPUTS: FF AND ARRAY FC$
3ØØ6 REM OUTPUTS: DATA CODES IN FC$(Ø),...,FC$(FF-1), AND NC,CW
3ØØ8 REM OVERWRITTEN VARIABLES: I1$,NC,CW
3Ø1Ø REM
3Ø12 NC=Ø
3Ø14 PRINT
3Ø16 PRINT "ENTER DATA CODE(S) ONE PER LINE, TERMINATE WITH -1 CODE."
3Ø18 INPUT I1$
3Ø2Ø IF I1$="-1" THEN 3Ø3Ø
3Ø22 FC$(NC)=I1$
3Ø24 NC=NC+1
3Ø26 IF NC>=FF THEN 3Ø3Ø
3Ø28 GOTO 3Ø18
3Ø3Ø PRINT "ENTER PERCENTAGE COSINE TAPER TO BE USED."
3Ø32 INPUT CW
3Ø34 IF CW>5Ø THEN CW=5Ø
3Ø36 IF CW<Ø THEN CW=Ø
3Ø38 RETURN
3Ø4Ø REM **** END ROUTINE TO INPUT DATA CODES ****
3Ø42 REM
31ØØ REM **** ROUTINE TO READ HEADER FILE, SET UP SCRATCH WAVEFORM ****
31Ø2 REM
31Ø4 REM INPUTS: FO$ (FOUR CHARACTER 'MDP' DATA FILE NAME)
31Ø6 REM OUTPUTS: ID$,ST$,FN$,NB,DT,F,ND,N,HU$,VU$
31Ø8 REM OVERWRITTEN VARIABLES: I1$ AND OUTPUTS
311Ø REM
```



@

```
3112 CLOSE #1
3114 I1$=FO$&"Ø.DAT"
3116 OPEN #1 AS I1$ FOR READ
3118 READ #1,ID$,ST$,FN$
312Ø READ #1,NB,DT,F,ND,IB,IE
3122 CLOSE #1
3124 N=IE-IB+1
3126 REM
3128 DELETE WR,RY
313Ø WAVEFORM WR IS RY(N-1),DT,HU$,VU$
3132 RETURN
3134 REM **** END ROUTINE TO READ HEADER FILE, SET UP WAVEFORM ****
3136 REM
32ØØ REM **** ROUTINE TO READ IN A DATA SET ****
32Ø2 REM
32Ø4 REM INPUTS: FO$;CURRENT DATA CODE IN CC$;SCRATCH WAVEFORM WR
32Ø6 REM OUTPUTS: LABEL FOR DATA LB$;DATA IN WAVEFORM WR
32Ø8 REM OVERWRITTEN VARIABLES: LB$,WR,I1$
321Ø REM
3212 I1$=FO$&CC$&".DAT"
3214 OPEN #1 AS I1$ FOR READ
3216 READ #1,LB$
3218 READ #1,WR
322Ø CLOSE #1
3222 RETURN
3224 REM **** END ROUTINE TO READ IN A DATA SET ****
3226 REM
33ØØ REM **** ROUTINE TO CALCULATE AMPLITUDE SPECTRUM ****
33Ø2 REM
33Ø4 REM INPUTS: N;CW, THE PERCENTAGE COSINE TAPER;
33Ø6 REM WR (RY,DT,HU$,VU$), CONTAINING THE DATA;
33Ø8 REM SCRATCH WAVEFORM W1 (A1,D1,H1$,V1$) OF LENGTH N/2+1
331Ø REM OUTPUT: WAVEFORM W1 CONTAINING THE AMPLITUDE SPECTRUM (IN DB'S)
3312 REM NEW OR OVERWRITTEN VARIABLES: MN,RY,IK,AR,WF,W1,I,IX,ZI
3314 REM
3316 REM REMOVE THE MEAN, POSSIBLY APPLY THE COSINE WINDOW
3318 MN=MEA(RY)
332Ø RY=RY-MN
3322 IF CW=Ø THEN 3344
3324 IK=ITP(CW/1ØØ*N)
3326 IF IK=Ø THEN 3344
3328 AR=3.141593/IK
```

```
3330 FOR I=0 TO IK-1
3332 WF=.5*(1-COS(AR*I))
3334 RY(I)=WF*RY(I)
3336 RY(N-1-I)=WF*RY(N-1-I)
3338 NEXT I
3340 REM
3342 REM TRANSFORM, COMPUTE AMPLITUDE SPECTRUM (RELATIVE SCALE)
3344 RFFT1 WR
3346 A1(N/2)=RY(1)*RY(1)
3348 A1(0)=RY(0)*RY(0)
3350 FOR I=1 TO N/2-1
3352 A1(I)=RY(2*I)*RY(2*I)+RY(2*I+1)*RY(2*I+1)
3354 NEXT I
3356 AR=MAX(A1)
3358 A1=A1/AR
3360 REM AVOID LOG OF ZERO PROBLEM BY ADDING A "SMALL" NUMBER
3362 REM IF NECESSARY
3364 REM
3366 IF MIN(A1)>0 THEN 3388
3368 REM SET ZEROS TO -144 DB RELATIVE TO MAX. FREQ. COMPONENT
3370 AR=1/(2^48)
3372 IX=0
3374 ZI=CRS(A1(IX),0)
3376 IF ZI<0 THEN 3388
3378 A1(ZI)=AR
3380 IX=ZI+1
3382 IF IX>N/2 THEN 3388
3384 GOTO 3374
3386 REM
3388 AR=10/LOG(10)
3390 A1=AR*LOG(A1)
3392 D1=DT
3394 H1$=HU$
3396 V1$='DB'
3398 RETURN
3400 REM **** END OF ROUTINE TO CALCULATE AMPLITUDE SPECTRUM ****
3402 REM
```

```
2500 REM **** MDPA3.BAS ROUTINE TO ESTIMATE AND DISPLAY RMS TIME  ****
2502 REM **** JITTER AND THE RMS RESIDUAL QUANTIZATION ERROR FROM ****
2504 REM **** THE ANALOG ERRORS OF A DIGITIZED SINUSOIDAL SIGNAL  ****
2508 REM
2510 REM INPUTS:  FOUR CHARACTER DATA FILE NAME FO$
2512 REM OUTPUTS:  PRINTED RMS ERROR ESTIMATES
2514 REM NEW OR OVERWRITTEN VARIABLES: I1$,ID$,ST$,FN$,PA,NB,DT,F,ND,IB
2516 REM IE,N,RY,HU$,VU$,LB$,WR,NU,ER,MN,FA,J,SM,XX,XY,XM,YM,TJ,TE,QE,JU
2518 REM
2520 REM READ IN HEADER FILE, CHECK FOR NONSINUSOIDAL SIGNAL
2522 REM
2550 PAGE
2552 CLOSE #1
2554 I1$=FO$&"0.DAT"
2556 OPEN #1 AS I1$ FOR READ
2558 READ #1,ID$,ST$,FN$
2560 IF ST$='SINE' THEN 2570
2562 PRINT "ERROR: DIGITIZED SIGNAL IS NOT SINUSOIDAL."
2564 PRINT "^G^G"
2566 WAIT
2568 GOTO 2830
2570 DELETE PA,WR,RY,WE,ER,MN,NU
2572 DIM PA(3)
2574 READ #1,NB,DT,F,ND,IB,IE,PA
2576 CLOSE #1
2578 N=IE-IB+1
2580 REM
2582 REM SET UP SCRATCH WAVEFORM AND READ ANALOG ERRORS
2584 WAVEFORM WR IS RY(N-1),DT,HU$,VU$
2586 I1$=FO$&"2.DAT"
2588 OPEN #1 AS I1$ FOR READ
2590 READ #1,LB$,WR
2592 CLOSE #1
2594 REM
2596 REM CONSTRUCT ERROR VERSUS SLOPE^2 ARRAY
2598 REM
2600 NP=5*ITP(N/50+.5)
2602 IF NP<5 THEN NP=5
2604 DIM NU(NP)
2606 WAVEFORM WE IS ER(NP),DT,HU$,VU$
2608 PI=3.141593
2610 SM=2*PI*PA(1)*PA(0)
```

```
2612 FA=2*PI*PA(1)*DT
2614 FOR J=Ø TO N-1
2616 TJ=COS(FA*(J+IB)+PA(2))
2618 TJ=ITP(TJ*TJ*NP+.5)
262Ø NU(TJ)=NU(TJ)+1
2622 ER(TJ)=ER(TJ)+RY(J)*RY(J)
2624 NEXT J
2626 REM
2628 REM FIT ERROR MODEL TO ERROR DATA
263Ø REM
2632 QE=RMS(RY)
2634 DELETE RY
2636 DIM RY(NP)
2638 XM=Ø
264Ø YM=Ø
2642 XX=Ø
2644 XY=Ø
2646 YY=Ø
2648 K=Ø
265Ø FOR J=Ø TO NP
2652 RY(J)=J
2654 IF NU(J)=Ø THEN 2664
2656 K=K+1
2658 ER(J)=ER(J)/NU(J)
266Ø XM=XM+J
2662 YM=YM+ER(J)
2664 NEXT J
2666 XM=XM/K
2668 YM=YM/K
267Ø FOR J=Ø TO NP
2672 IF NU(J)=Ø THEN 268Ø
2674 XX=XX+(J-XM)*(J-XM)
2676 XY=XY+(J-XM)*(ER(J)-YM)
2678 YY=YY+(ER(J)-YM)*(ER(J)-YM)
268Ø NEXT J
2682 TJ=XY/XX
2684 SS=(YY-TJ*TJ*XM)/(K-2)
2686 TE=SQR(SS*NP*NP/XX)/(SM*SM)
2688 QR=SQR((1/K+XM*XM/XX)*SS)
269Ø JL=1/SM
2692 JU=1/(PI*PA(1)*SQR(2*PA(Ø)))
2694 REM
```



@

```
2696 REM PRINT INTERVAL OF RELIABLE JITTER ESTIMATES
2698 REM
2700 PRINT "INTERVAL OF RELIABLE TIME JITTER ESTIMATES:"
2702 PRINT "   ";JL;"  TO ";JU;" SECOND."
2704 PRINT
2706 PRINT "THE LOWER BOUND IS REQUIRED TO RESOLVE JITTER AMPLITUDE"
2708 PRINT "EFFECTS AND THE UPPER BOUND TO INSURE THAT THE MODELING"
2710 PRINT "USED IS VALID."
2712 PRINT
2714 REM
2716 REM ESTIMATE RMS JITTER AND RMS ADDITIVE ERROR, TEST FOR
2718 REM THEIR SIGNIFICANCE, PRINT RESULTS.
2720 REM
2722 IF TJ>0 THEN 2734
2724 TJ=0
2726 PRINT "NO TIME JITTER DETECTED."
2728 PRINT
2730 PRINT "RMS ADDITIVE ERROR = ";QE;" LSB."
2732 GOTO 2790
2734 IF YM-TJ*XM>=0 THEN 2742
2736 QE=0
2738 QR=-1
2740 GOTO 2744
2742 QE=SQR(YM-TJ*XM)
2744 TJ=SQR(TJ*NP)/SM
2746 PRINT "MEAN SQUARE TIME JITTER ESTIMATE = ";TJ*TJ;" SECOND^2."
2748 PRINT "                  STANDARD ERROR = ";TE;" SECOND^2."
2750 PRINT "ESTIMATE IS ";
2752 IF TJ*TJ<=1.645*TE THEN PRINT "NOT ";
2754 PRINT "SIGNIFICANT AT THE APPROXIMATE 95% LEVEL."
2756 PRINT
2758 PRINT "RMS TIME JITTER ESTIMATE = ";TJ;" SECOND."
2760 PRINT
2762 PRINT
2764 IF QR>-1 THEN 2770
2766 PRINT "NO RESIDUAL ADDITIVE ERROR DETECTED."
2768 GOTO 2790
2770 PRINT "MEAN SQUARE RESIDUAL ADDITIVE ERROR = ";QE*QE;" LSB^2."
2772 PRINT "                     STANDARD ERROR = ";QR;" LSB^2."
2774 PRINT "ESTIMATE IS ";
2776 IF QE*QE<=1.645*QR THEN PRINT "NOT ";
2778 PRINT "SIGNIFICANT AT THE APPROXIMATE 95% LEVEL."
```

```
2780 PRINT
2782 PRINT "RMS RESIDUAL ADDITIVE ERROR ESTIMATE = ";QE;" LSB."
2784 REM
2786 REM CONSTRUCT FITTED ERROR ARRAY
2788 REM
2790 IF TJ>0 THEN 2796
2792 RY=QE*QE
2794 GOTO 2798
2796 RY=YM+TJ*TJ*SM*SM/NP*(RY-XM)
2798 DT=SM/NP
2800 HU$="(LSB/"&HU$&")^2"
2802 VU$="LSB^2"
2804 REM
2806 REM GRAPH ERROR VS. INPUT SLOPE^2 AND FITTED MODEL.
2808 REM
2810 PRINT "^G^G"
2812 WAIT
2814 PAGE
2816 PRINT "MEAN SQUARE ANALOG ERROR VERSUS INPUT SLOPE^2"
2818 PRINT "(DOTS ARE VALUES PREDICTED BY FITTED MODEL):"
2820 GRAPH WE
2822 DISPLAY 1,WR
2824 PRINT "^G^G"
2826 WAIT
2828 DELETE WR,RY,WE,ER,MN,NU
2830 RETURN
2832 REM **** END ROUTINE MDPA3.BAS ****
2834 REM
```



@

```
2500 REM **** MDPA4.BAS ROUTINE TO CALCULATE AND GRAPH RMS ****
2502 REM **** ANALOG ERROR VERSUS DIGITIZER OUTPUT CODE.   ****
2504 REM
2506 REM INPUT: FOUR CHARACTER DATA FILE NAME FO$
2508 REM OUTPUTS: GRAPH ONLY
2510 REM NEW OR OVERWRITTEN VARIABLES: I1$,I2$,ID$,ST$,FN$,NB
2512 REM    DT,F,ND,IB,IE,NC,N,A1,DD,ER,W1,D1,H1$,V1$,,NR,LN,LB$
2514 REM     K,QQ,DC
2516 REM
2518 REM READ IN HEADER FILE, SET UP ARRAYS
2520 REM
2550 RELEASE ALL
2552 CLOSE #1
2554 I1$=FO$&"0.DAT"
2556 OPEN #1 AS I1$ FOR READ
2558 READ #1,ID$,ST$,FN$
2560 READ #1,NB,DT,F,ND,IB,IE
2562 CLOSE #1
2564 NC=2^NB
2566 N=IE-IB+1
2568 DELETE IN,W1,A1,DD,ER
2570 INTEGER IN(NC-1)
2572 WAVEFORM W1 IS A1(NC-1),D1,H1$,V1$
2574 D1=1
2576 H1$="LSB"
2578 V1$="LSB"
2580 REM
2582 REM READ IN A MAXIMUM OF 512 POINTS AT A TIME
2584 NR=N/512
2586 IF NR>1 THEN 2594
2588 NR=1
2590 LN=N
2592 GOTO 2596
2594 LN=512
2596 DIM DD(LN-1),ER(LN-1)
2598 REM
2600 REM OPEN DATA FILES AND CONSTRUCT ARRAYS OF DIGITIZER ERROR SUM OF
2602 REM SQUARES VERSUS DIGITAL CODE, AND NO. PER DIGITAL CODE
2604 REM
2606 I1$=FO$&"1.DAT"
2608 I2$=FO$&"2.DAT"
2610 CLOSE #2
```

```
2612 OPEN #1 AS I1$ FOR READ
2614 OPEN #2 AS I2$ FOR READ
2616 READ #1,LB$
2618 READ #2,LB$
2620 FOR QQ=1 TO NR
2622 READ #1,DD
2624 READ #2,ER
2626 FOR K=0 TO LN-1
2628 DC=DD(K)
2630 A1(DC)=A1(DC)+ER(K)*ER(K)
2632 IN(DC)=IN(DC)+1
2634 NEXT K
2636 NEXT QQ
2638 CLOSE #1
2640 CLOSE #2
2642 REM
2644 REM CALCULATE RMS ERROR PER DIGITAL CODE
2646 REM
2648 FOR K=0 TO NC-1
2650 IF IN(K)=0 THEN 2654
2652 A1(K)=A1(K)/IN(K)
2654 NEXT K
2656 A1=SQR(A1)
2658 DELETE IN,ER,DD
2660 REM
2662 REM GRAPH RESULTS
2664 REM
2666 PAGE
2668 PRINT "RMS ANALOG ERROR VERSUS DIGITIZER OUTPUT CODE:"
2670 GRAPH W1
2672 PRINT "^G^G"
2674 WAIT
2676 DELETE W1,A1
2678 RETURN
2680 REM **** END ROUTINE MDPA4.BAS ****
2682 REM
```

```
2500 REM **** MDPA5.BAS ROUTINE TO CALCULATE AND GRAPH ****
2502 REM **** DIGITIZER RMS ANALOG ERROR VERSUS PHASE  ****
2504 REM
2506 REM INPUTS:  FOUR CHARACTER DATA FILE NAME FO$
2508 REM OUTPUTS: GRAPH ONLY
2510 REM NEW OR OVERWRITTEN VARIABLES: I1$,ID$,ST$,FN$,PA,NB,DT,F,ND
2512 REM  IB,IE,LB$,WR,RY,HU$,NU,ER,WE,DE,PI,FA,PH,J,NP
2514 REM
2516 REM READ IN HEADER FILE, CHECK FOR NONSINUSOIDAL SIGNAL
2518 REM
2550 PAGE
2552 CLOSE #1
2554 I1$=FO$&"0.DAT"
2556 OPEN #1 AS I1$ FOR READ
2558 READ #1,ID$,ST$,FN$
2560 IF ST$='SINE' THEN 2570
2562 PRINT "ERROR: DIGITIZED SIGNAL IS NOT SINUSOIDAL."
2564 PRINT "^G^G"
2566 WAIT
2568 GOTO 2656
2570 DELETE PA,WR,RY,WE,ER,NU
2572 DIM PA(3)
2574 READ #1,NB,DT,F,ND,IB,IE,PA
2576 CLOSE #1
2578 N=IE-IB+1
2580 REM
2582 REM SET UP SCRATCH WAVEFORM AND READ ANALOG ERRORS
2584 REM
2586 WAVEFORM WR IS RY(N-1),DT,HU$,VU$
2588 I1$=FO$&"2.DAT"
2590 OPEN #1 AS I1$ FOR READ
2592 READ #1,LB$,WR
2594 CLOSE #1
2596 REM
2598 REM CONSTRUCT RMS ERROR VERSUS PHASE ARRAY
2600 REM
2602 NP=5*ITP(N/50+.5)
2604 IF NP<5 THEN NP=5
2606 DIM NU(NP-1)
2608 WAVEFORM WE IS ER(NP-1),DE,HU$,VU$
2610 PI=3.141593
2612 FA=2*PI*PA(1)*DT
```

```
2614 FOR J=0 TO N-1
2616 PH=(FA*(J+IB)+PA(2))/(2*PI)+1/(2*NP)
2618 PH=ITP((PH-ITP(PH))*NP)
2620 IF PH<0 THEN PH=PH+NP
2622 NU(PH)=NU(PH)+1
2624 ER(PH)=ER(PH)+RY(J)*RY(J)
2626 NEXT J
2628 FOR J=0 TO NP-1
2630 IF NU(J)=0 THEN 2634
2632 ER(J)=SQR(ER(J)/NU(J))
2634 NEXT J
2636 DE=2/NP
2638 HU$='PI'
2640 REM
2642 REM GRAPH RESULTS
2644 REM
2646 PRINT "RMS ANALOG ERROR VERSUS PHASE:"
2648 GRAPH WE
2650 PRINT "^G^G"
2652 WAIT
2654 DELETE NU,WE,ER
2656 RETURN
2658 REM **** END OF ROUTINE MDPA5.BAS ****
2660 REM
```

**PROGRAM NAMES:** MDPMPN, MDPANL

**AUTHORS:** Lyle Ochs, Paul McClellan
Signal Analysis Group
94-384

**DATE:** November 10, 1977

**ABSTRACT:**

MDPMPN and MDPANL are TEK SPS BASIC programs used to digitally measure the dynamic performance of digitizers. Program MDPMPN first synthesizes a digitizer's analog input by least squares fitting an appropriate model to the digitizer output. It then calculates summary statistics describing the performance of the digitizer and, if desired, stores data for later analysis by program MDPANL. This program supports a variety of analysis options that may be applied to the digitizer data and other data generated by MDPMPN.

**DISCUSSION:**

Historically, A/D and digitizer specifications have been of the DC or static type and have often been unclear or incomplete. It is sometimes desirable to augment these static specifications with actual measurements of digitizer performance.

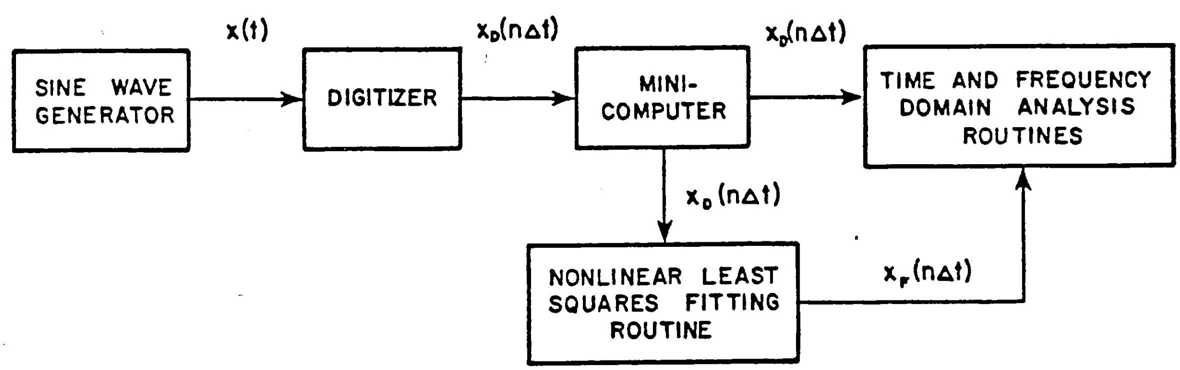
In the past, test setups for measuring digitizer performance have involved specialized analog instrumentation and reference signals. However, valid performance specifications result only if the errors contributed by the instrumentation are small compared to those of the A/D under test. This can be an especially troublesome problem in testing the higher resolution, higher speed A/D converters available today.

A direct procedure for evaluating digitizer performance is to interface the instrument to a computer and analyze the output in its own domain -- digitally. In this procedure, the test setup consists of a signal generator, digitizer, minicomputer, graphics terminal, and a set of analysis routines. Typical choices for the signal generator include precision voltage sources for static testing and sine wave generators or random noise generators for dynamic tests.

In this digital approach, there are no extraneous error components from D/A's, comparators, scopes, etc. found in test setups employing analog instrumentation for measurement and analysis. But specifications for the reference signal are critical in this direct, digital test setup, and they must not exceed theoretical accuracy, resolution, and noise limits of the digitizer to be tested. For sine wave testing, the insertion of a bandpass filter between the signal generator and digitizer can substantially reduce these problems by rejecting harmonic distortion components and reducing noise bandwidth.

Focusing on dynamic testing with sine waves for reference signals, a time domain approach can be taken to analyze digitizer performance. This method involves synthesizing the analog input (reference) signal by fitting a sine wave model to the digitizer output via nonlinear least squares. It allows for both time and frequency domain analysis.

Programs MDPMPN and MDPANL have been developed to analyze the digitizer output data using this time domain approach. Program MDPMPN synthesizes the analog input and calculates some measures of digitizer performance. If desired, it will also save data for further analysis by program MDPANL. The diagram outlines the technique.




In this approach, the analog input signal is a sine wave represented mathematically by the equation:

$$X(t) = A \sin (2\pi ft+\Theta) + C,$$

where A, f, $\Theta$, and C are the amplitude, frequency, phase, and DC offset parameters. The digitizer output is $X_D$ $(n\Delta t)$, where $\Delta t$ is the sampling interval, and n is the sample index. MDPMPN requires that the digitizer output be stored in a data file. The program reads the data and overlays a subroutine to synthesize the analog input. In this subroutine the digitizer output and the sine wave frequency are first used to form initial estimates of the remaining parameters A, $\Theta$, and C. All this data is then used in a nonlinear least squares fitting routine which determines the fitted parameter values.

MDPMPN next overlays a subroutine to calculate summary statistics. The model with fitted parameters is:

$$X_F(t) = \hat{A} \sin (2\pi\hat{f}t+\hat{\Theta}) + \hat{C}.$$

This is taken to be the analog input to the digitizer. $X_F(t)$ is "perfectly sampled" by computer subroutine, producing $X_F(n\Delta t)$, and this perfectly sampled analog input is subtracted, sample by sample, from the digitizer output to form the analog error signal

$$r(n\Delta t) = X_D(n\Delta t) - X_F(n\Delta t).$$

The minimum, maximum, RMS analog error, and signal-to-noise ratio are calculated and printed. The signal-to-noise ratio is the ratio of RMS synthesized input (with mean removed) to RMS analog error, expressed in dB's.

The summary statistics subroutine also "perfectly quantizes" the sampled input data $X_F(n\Delta t)$ -- call this data $X_{FD}(n\Delta t)$ -- and the analog error signal for an ideal digitizer with the same number of bits as the digitizer under test is calculated. In equation form, the errors are

$$R(n\Delta t) = X_{FD}(n\Delta t) - X_F(n\Delta t).$$

The ratio RMS(r)/RMS(R) provides a measure of actual digitizer performance relative to ideal performance. Taking the base two logarithm gives "lost bits". For a b bit digitizer, the effective bits for the digitized data $X_D$ is then

$$EB(X_D) = b - \log_2(RMS(r)/RMS(R)),$$

which is also calculated and printed. Note that effective bits may decrease due to either systematic or random errors in the digitization process since either type of error will tend to inflate RMS(r).

The above least squares fitting procedure and summary statistics calculations are automatically made by program MDPMPN. If further performance measurements are desired, the program will save the digitized output, synthesized input, analog errors, ideally digitized output, and ideal digitizer analog errors on data files for further analysis by program MDPANL. The analysis options available are 1) graphing one or two data sets, 2) computing and graphing amplitude spectra of one or two data sets, 3) estimating RMS time jitter and RMS residual additive error for sine wave input signals, 4) calculating and graphing RMS analog error versus digital code, and 5) calculating and graphing RMS analog error versus phase of sine wave input signals.

Time jitter is defined for our purposes as the time uncertainty in the points at which the input signal was sampled and quantized. The signal at the time jittered point $n\Delta t+e_n$ may be approximated by the first order Taylor series

$$X(n\Delta t+e_n) \doteq X(n\Delta t) + e_n X'(n\Delta t).$$

The amplitude uncertainty due to the time jitter $e_n$ is then

$$X(n\Delta t+e_n) - X(n\Delta t) = e_n X'(n\Delta t),$$

so the effect of time jitter in the digitizer output is an amplitude uncertainty that is proportional to the slope of the input signal at each sample point. This uncertainty is in addition to the digitizer's quantization errors, which are independent of the slope of the input signal (except when coherently sampling). Hence to detect and estimate time jitter from digitizer output, slope dependent analog errors need to be detected and measured.

To do this, the following mean square analog error model is used:

$$E_t^2(s^2) = E_r^2 + E_j^2\, s^2,$$

where $E_t^2(s^2)$ is the total mean square analog error as a function of squared input slope, $s^2$, $E_j^2$ is the mean square time jitter, and $E_r^2$ is the residual mean square analog error, independent of the input slope s. The input slope is calculated and sampled using the fitted input signal parameters:

$$s(n\Delta t) = X_F'(n\Delta t) = 2\pi\hat{A}\hat{f}\sin(2\pi\hat{f}n\Delta t + \hat{\theta})$$

For sine wave inputs, the range of $s^2$ values is divided into intervals. The analog errors, $r(n\Delta t)$, are grouped according to these $s^2$ intervals, and for each interval the mean square analog error, $E^2{}_t(s^2)$, is calculated. In this way, the independent and dependent variables for fitting a straight line (simple linear regression) are constructed. For some values of $s^2$ there may be no observations, and these $s^2$ values are then ignored in the remaining calculations. $E_r^2$ and $E_j^2$ are then estimated by fitting a straight line to this data by least squares, giving $\hat{E}_r^2$ and $\hat{E}_j^2$. If $\hat{E}_j^2 \leq 0$, then time jitter has not been detected and we set $\hat{E}_j^2 = 0$, $\hat{E}_r^2 =$ mean square analog error. Otherwise, $\hat{E}_j^2$ is tested for statistical significance and RMS time jitter calculated. $\hat{E}_r^2$ is also tested for significance and RMS residual error calculated.

The above analog error model is valid only for a range of time jitter values, depending on the sine wave input used. If $E_j^2$ is too small, the amplitude effects of time jitter will be buried in quantization noise. If it is too large, additional, higher order terms would need to be included in the error model. For an input sine wave signal described by $A\sin(2\pi ft + \theta) + C$, where A is expressed in LSBs, the interval over which the model is valid is:

$$1/(2\pi A) < E_j f < 1/(\pi\sqrt{2A})$$

MDPANL calculates this interval along with the above estimates.

It is important to keep in mind that the time jitter estimation procedure does not directly measure time jitter. Rather, it looks for a component of mean square analog error that is linearly related to the squared input slope in a positive sense. If this dependence is detected, the amount of time jitter that could have produced the same amount of linear dependence is then estimated. If the time jitter lies within the interval defined above, the time jitter should be detected in this way. However, if systematic errors occur, especially if they are near the sine wave "zero" crossings, this procedure may detect significant "time jitter" when none or an insignificant amount is really present. Thus the time jitter estimate should not be viewed in isolation from a graph of the analog errors and other analyses.

By systematically varying the frequency of the input sine wave signal and repeating some or all of the analyses provided by programs MDPMPN and MDPANL for each frequency, dynamic performance versus frequency can be graphically displayed.

The above programs may also use ramp or DC input signals to help evaluate digitizer performance. Ramp signals can be used to measure digitizer dynamic linearity at varying ramp rates. Some digitizers may have static nonlinearities across their record lengths that can be most easily seen in acquired DC signals. However, not all of the analyses performed on digitized sine wave data can be performed on digitized ramp or DC data. Also, it is generally more difficult to evaluate the accuracy of a ramp signal generator than a sine wave signal generator.

Programs MDPMPN and MDPANL can be used in measuring the dynamic performance of digitizers. However, static, or DC, performance measurements may also be desired. These measurements can be made using program ADSPAN, documented elsewhere.

## OPERATING ENVIRONMENT AND INSTRUCTIONS:

### A. Signal Generator Requirements

Any sine wave, ramp, or DC generator used to supply test signals should generate signals accurate to within the resolution of the ideally digitized signals.

A candidate sine wave generator can be evaluated using a spectrum analyzer. The total harmonic distortion and noise level should be less than the total harmonic distortion of a sine wave digitized by an ideal digitizer with the same number of bits as the digitizer under evaluation. For an ideal b-bit quantizer, the total harmonic distortion is THD $\doteq - 6.02\ b - 1.76$ dB's. If the spectrum analyzer does not have sufficient dynamic range to verify that the signal generator satisfies this requirement, a bandpass filter can be inserted between the signal generator and digitizer to reduce what harmonic distortion is present to be sure it is satisfied. This will also reduce the noise bandwidth.

Also, the frequency of the sine wave generator output should be stable over the digitized record length. Frequency jitter can be detected on a spectrum analyzer as a widening of the fundamental peak. For a given sine wave frequency f, the allowable short-term frequency jitter $\Delta f$ is

$$|\Delta f/f| \leq e/(2^b \pi n),$$

where n is the number of cycles in the record to be digitized, b is the number of bits in the digitizer output words, and e is the acceptable amplitude uncertainty in LSBs. A reasonable choice for e is 1/4 LSB. For example, suppose 100 cycles of a 20 megahertz sine wave is to be acquired on an 8-bit digitizer, and e = 1/4. Then $|\Delta f| \leq 62.2$ hertz. If e is expressed as a RMS value, $\Delta f$ is also. For lower frequencies, many cycles, and many bits, this may be a difficult performance requirement to verify.

Ramp signal generators can be difficult to evaluate. A candidate ramp generator's output should be linear to well within the resolution of the digitizer under test, say 1/4 LSB or less. If another digitzer with greater resolution than the one under evaluation is available (say having two or more additional bits) the ramp signal could be digitized with it and its performance verified. Otherwise, the signal might be passed through a differentiator and the result evaluated as an ideal DC signal. However, the differentiator may have its own nonlinearities, distorting the measurement.

A DC signal might also be evaluated by digitizing it with a digitizer with higher resolution than the digitizer under evaluation. Or its AC component may be viewed on a sufficiently sensitive oscilloscope. However, in either of these methods the nonlinearities of the measuring instrument should be adjusted for.

## B. Program Operating Instructions.

Data acquisition and analysis is divided into three conceptual phases, each requiring a BASIC program to be OLDed and RUN.

1) **Data Acquisition** (user supplied program). This is the transfer of data from the local memory of the digitizer under test to an ASCII file required by the "MDP" programs. The data should be previewed at this time and the ASCII file must have a form defined by the following PRINT statements:

```
PRINT #1,ID$
PRINT #1,ST$
PRINT #1,NB,DT,F,ND,IB,IE
FOR I=0 TO ND-1
PRINT #1,Y(I)
NEXT I
```

where:

ID$--An identification string for the digitizer data, e.g., date, instrument, etc.

ST$--A string designating the signal type. ST$ may be DC, RAMP, or SINE currently. These are the only signal types that the MDP routines know how to fit.

NB---The number of bits in the digitizer words.

DT---The sampling interval in seconds.

F----If the signal type is SINE, this variable should be set to its frequency. For other signal types its value is ignored.

ND---The number of acquired data points, Y(0),Y(1),...,Y(ND-1).

IB,IE--The starting and ending indices of the data to be analyzed. Typically IB=0 and IE=ND-1.

Y(0),Y(1),...,Y(ND-1)--The ND NB-bit digitizer words. Y should be an integer array.

The TEK SPS BASIC routine to transfer digitizer output from local memory to computer memory, preview the data, and write out the raw data file is provided by the user.

2) **Fitting Data, Generating Summary Statistics and Intermediate Data Files** (program MDPMPN). Once the raw data is on file, a fitting routine can be called to construct a computer synthesized version of the analog signal that was input to the digitizer. Summary statistics are then printed and intermediate data files can be created for later and more complete analysis.

To complete the above steps, simply OLD the program on file MDPMPN.BAS and RUN it. It first asks for the raw data file name, reads the information on it, and graphs the data points Y(IB),Y(IB+1),...,Y(IE) to be analyzed. The program will next ask for any desired changes in IB and IE and then overlay the appropriate least squares fitting routine (MDPRMP.BNR for DC or RAMP signals or MDPSIN.BNR for SINE signals). After the analog input has been synthesized, the program finally overlays and calls MDPSS.BNR. This routine asks for a file name for output of intermediate data, if desired, then computes and prints summary statistics, and, finally, writes out the intermediate data if that option was chosen.

The main program, MDPMPN.BAS, can be immediately rerun on the same or a new raw data file, unless a data record longer than 1024 points was analyzed. In that case, parts of the main program were deleted to provide needed additional memory space and MDPMPN.BAS will need to be OLDed again.

3) **Analysis of Intermediate Data** (program MDPANL). By OLDing and RUNning MDPANL.BAS, various analysis routines can be overlaid and the intermediate data read in and analyzed. MDPANL.BAS prints a "menu" of analysis options and data sets. Analysis routines are named MDPAXX.BNR, where XX is a one or two digit integer. The ".BNR" extension is used to designate a BASIC routine stripped of REMark statements. Currently five analysis options are available. These are:

MDPA1.BNR--graphs one or two data sets.

MDPA2.BNR--computes and graphs amplitude spectra of one or two data sets.

MDPA3.BNR--estimates RMS time jitter and RMS residual additive error for SINE type input signals.

MDPA4.BNR--calculates and graphs RMS analog error versus digital output code.

MDPA5.BNR--calculates and graphs RMS analog error versus phase of SINE type input signals.

Note that options 3 and 5 require SINE type input signals. Only analysis options 1 and 2 are valid for DC type signals, and only 1, 2, and 4 are valid for RAMP type signals.

Additional analysis routines can be written by the user; just follow the data reading, program line number, and house cleaning conventions of the above routines.

## C. Additional Program Notes:

1) If analysis option MDPA2.BNR or any other analysis using the FFT is to be used, the length of the data record to be analyzed, IE-IB+1, must be a power of 2.

2) A number of control parameters are set in the nonlinear least squares fitting routine used for SINE type signals (MDPSIN.BNR). For an explanation of these, a copy of MDPSIN.BAS (with REMarks) is available.

3) If a sawtooth signal is used to generate RAMP type input signals, only one ramp at a time should be fitted by MDPRMP.BNR.

4) Less than one cycle of a SINE type signal can be fitted by MDPSIN.BNR if the data points are not clustered around an extremum.

5) The units of the digitizer output are set to seconds ("S") and least significant bits ("LSB").

6) To speed up data storage and retrieval from raw data files and also to reduce raw data file lengths, replace PRINT's and INPUT's by WRITE's and READ's, respectively, when storing or retrieving raw data in the user supplied program and program MDPMPN.

7) The program files necessary for phases 2) and 3) above are:

MDPMPN.BAS
    MDPRMP.BNR
    MDPSIN.BNR
    MDPSS.BNR

MDPANL.BAS
    MENU.MDP (menu file, not a program)
    MDPA1.BNR
    MDPA2.BNR
    MDPA3.BNR
    MDPA4.BNR
    MDPA5.BNR
    .
    .
    .

Copies of the above routines with remarks (.BAS extension) are available.