

206  
9-29-80 lrc

Dec. 1989

SAND80-0881  
Unlimited Release  
UC-37

## A Two-Stage Nonrecursive Filter/Decimator

James R. Yoder, Billy D. Richard

**MASTER**



Sandia National Laboratories

A TWO-STAGE NONRECURSIVE FILTER/DECIMATOR

James R. Yoder  
Data Planning Division 2425

Billy D. Richard  
QA and Engineering Information Design Division 2655

Sandia National Laboratories  
Albuquerque, NM 87185

ABSTRACT

A two-stage digital filter/decimator has been designed and implemented to reduce the sampling rate associated with the long-term computer storage of certain digital waveforms. This report describes the design selection and implementation process and serves as documentation for the system actually installed. A filter design with finite-impulse response (nonrecursive) was chosen for implementation via direct convolution. A newly-developed system-test statistic validates the system under different computer-operating environments.

DISCLAIMER

The information contained herein is the property of Sandia National Laboratories. It is loaned to you for your internal use only. It is not to be distributed outside your organization. Sandia National Laboratories is not responsible for the accuracy or completeness of the information contained herein. Sandia National Laboratories is not responsible for the accuracy or completeness of the information contained herein. Sandia National Laboratories is not responsible for the accuracy or completeness of the information contained herein. Sandia National Laboratories is not responsible for the accuracy or completeness of the information contained herein.

Fig 3

#### ACKNOWLEDGMENTS

The completion of this project required the cooperative interaction of several groups: chiefly, the Generator Development, QA and Engineering Information Design, and Data Planning Divisions at Sandia National Laboratories Albuquerque, as well as the Business Information Systems and the Quality Audits, Statistics, and Customer Liaison Organization at the General Electric Neutron Devices plant (GEND) in St. Petersburg, Florida. We are especially indebted to J. E. Jackson and C. A. Wyatt of GEND.

S. D. Stearns (1111) of Sandia has contributed several essential ideas to this and related Digital Signal Processing endeavors. The authors are appreciative of his continuing support.

E. E. Ard from Sandia's Quality Assurance Advanced Planning Division (1417) and R. G. Easterling from the Statistics, Computing, and Human Factors Division (1223) were very helpful during the statistical evaluation portion of the project.

## CONTENTS

	<u>Page</u>
Introduction	7
Ideal Filter/Decimator	8
Digital Filter Design	9
IIR Filter Evaluation	10
FIR Filter Evaluation	11
Final Design and Implementation of the FIR Filter	13
<i>Experimental Results</i>	18
Implementation and Testing of the Filter/Decimator at GEND	22
Conclusions	22

## ILLUSTRATIONS

### Figure

1	Flow of Data From the Product Tester at GEND to the Long-Term Storage Site at SNLA	7
2	Sample-Rate Reduction: Signal Passed Through Ideal Low-Pass Filter to Avoid Aliasing Distortion	8
3	Recursive (Infinite Impulse Response) Filter	10
4	Two-Stage FIR Implementation	12
5	First- and Second-Stage Filter Characteristics	17
6	E-Statistic Summary for GEND Product-Test Data	19
7	Plots of Original Signal, Output Signal, and Corresponding Frequency Spectra of the Two-Stage Digital Filter	19
8	E-Statistic Summary of White-Noise Signals	20
9	Plots of Original White-Noise Signal, Output Signal, and Corresponding Frequency Spectra of the FIR Filter	20

## TABLES

### Table

1	Filter-Design Evaluation Results	12
2	Key Frequencies With Normalized Equivalents	13
3	Comparison of the Ripple in Both the Passband and Stop Band of FIR Digital Low-Pass Filters	14
4	Effects on the Values of the Energy Statistics Resulting From Filter Coefficient Modification	21

## A TWO-STAGE NONRECURSIVE FILTER/DECIMATOR

### Introduction

Digital waveforms are acquired during the production and shelf-life testing of neutron generators at the General Electric Neutron Devices (GEND) plant in St. Petersburg, Florida. Because long-term storage of waveforms recorded at the original sampling rate is both necessary and expensive, the process of sampling-rate reduction was proposed that decreases the sampling rate and allows for long-term storage of the waveforms in the Product Data System at Sandia National Laboratories Albuquerque. In this report, we describe the software associated with the sampling-rate reduction process.

Figure 1 shows the flow of data from the product tester to Sandia. Initial use of the data for "quick-look" analysis and identification and for characterization of test equipment or product anomalies dictate the high initial sampling rate ( $T_1$ ). A subsequent resampling is performed at a much lower rate (sampling interval  $T_2 = T_1/10$ ) so that selected waveforms may be sent to Sandia for analysis and detection of long-term trends in product performance.

The need for a lower sampling rate and, therefore, shorter data records is indicated by the nature of the historical data system and by the large quantity of waveforms that are stored. Data stored in such a system not only consume space on the computer storage media (e.g., magnetic tape), but also use a significant amount of computer time. The data must be read and reread during subsequent file updating and back-up procedures and for every retrieval of even a subset of the historical file. Any reduction in record length consistent with the record content is economically rewarding.

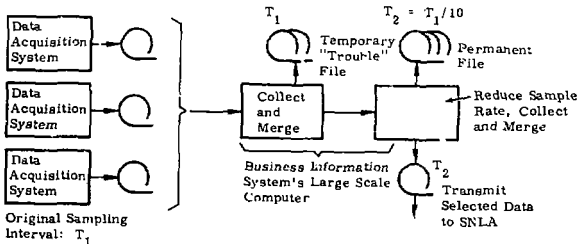


Figure 1. Flow of Data From the Product Tester at GEND to the Long-Term Storage Site at SLA

The process used for the sample-rate reduction is critical. The original signal must be band-limited in order to minimize the occurrence of digital aliasing in the signal that is resampled at the lower rate. Should the original signal not be sufficiently band-limited for the resampling process, it is necessary to filter the signal prior to sample-rate reduction. The effects of an improper sample-rate reduction process are discussed theoretically and graphically by Yoder,<sup>1</sup> and others,<sup>2,3,4</sup> This report will not review the theory nor expand the discussion of effects.

The sampling-rate reduction is achieved by a process known as decimation whereby only one out of M samples is saved and the rest are discarded. In the present case, a reduction of 1 in 10 is required (i. e., for every 10 samples, 1 is saved). The original digital waveform is 2000 samples long, but the last 500 samples are superfluous so that a truncation to 1500 samples is possible. From these 1500 samples, we want to save 150 regularly spaced samples, thereby achieving a 90% reduction in record length.

#### Ideal Filter/Decimator

Conceptually, the decimation process, by itself, is straightforward. However, the need to band-limit the input signals presents a somewhat more complex problem. Although the theoretical specifications for the band-limiting filter are easy to state, actual digital filter design and realization are somewhat difficult. Furthermore, the constraints imposed by the availability of a limited amount of computing time dictate the need for optimal processor designs, leading to a concatenated or staged filter/decimator process.

We begin by stating the requirements for an ideal antialiasing filter. We define:

$T_1$  = sampling interval associated with the original signal

$T_2$  = desired sampling interval for the resampled signal

M = decimation factor;  $T_2 = MT_1$

$\Omega$  = band limit associated with the original sequence

In general, aliasing will occur from sample-rate reduction unless  $T_1 \leq \frac{\pi}{M\Omega}$ . If this condition is not satisfied (as in the present case), digital-aliasing distortion can be avoided only by passing the original signal through an ideal low-pass filter with unit gain and cut-off frequency  $\pi/T_2$  (rad) or  $1/2T_2$  (Hz) (Figure 2).

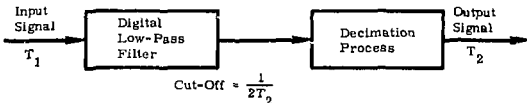


Figure 2. Sample-Rate Reduction: Signal Passed Through Ideal Low-Pass Filter to Avoid Aliasing Distortion

Added to the general criteria is the requirement that the filter operate in minimal time with at least linear, but preferably zero, phase distortion. Of course, the effects of finite word length upon parameter quantization must be considered. Finally, we must have some means for verifying that the system is operating correctly. Since the system will be installed in a computer-production environment, one can expect that either the computer hardware, software, or both will change. Therefore, we require that a system-performance test be available to assure us that a change to the computing environment results in minimal adverse change to the filter/decimator performance. To this end, we propose to examine a test statistic--the ratio of input to output energy for frequencies up to the ultimate Nyquist frequency (i.e., the highest resolvable frequency at the reduced sampling rate).

### Digital Filter Design

The input sampling rate is  $T_1$ ; we assume that this rate is sufficient to resolve all frequencies in the input waveform (since sufficient information to analyze the original sampling rate is unavailable). The Nyquist frequency for the incoming signal is then  $1/2T_1$ . We want a sample rate reduction of 10 which results in an ultimate Nyquist frequency of  $(1/10)(1/2T_1) = 1/20T_1 = 1/2T_2$  Hz.\* The low-pass filter must then have a nominal cut-off frequency of  $1/2T_2$  Hz.

There are two types of filters that can be applied: recursive (Infinite Impulse Response, IIR) and nonrecursive (Finite Impulse Response, FIR). Usually, the IIR filter is the faster of the two since it can normally be realized with fewer coefficients and, thereby, fewer multiply operations. However, this design suffers from two difficulties: nonlinear phase distortion and some uncertainty with respect to stability. Both problems can be overcome. The phase distortion can be eliminated by an input time-reversal process in which we run the filter twice but subject the input signal to two time reversals; performing the filter algorithm twice doubles the computing time required. Stability can be assured with careful design and with treatment of parameter quantization effects.

FIR filters are, in general, slower than IIR filters because more coefficients are required; therefore more multiply operations are performed as part of the filter algorithm. However, FIR filters can be designed with linear (easily treatable) phase shifts. Certain procedures can also be implemented to increase the computation speed. The choice between an FIR and IIR filter design then depends upon the function it is required to perform with respect to a given application.

---

\*The actual sampling frequencies are classified and may be found in DR266301, Data Requirements Specification.

### IIR Filter Evaluation:

Several types of IIR filters are available, but we use a Butterworth design for its maximally flat passband feature. Based on Stearns' procedure,<sup>5</sup> we established the following specifications:

$$\text{Passband: } 0 - \frac{1}{20T_1}$$

$$\text{Skirt: } \text{Down 3 dB at } (0.8) \left( \frac{1}{20T_1} \right) = \frac{1}{25T_1}$$

$$\text{Stop Band: } \text{Down 60 dB at } \frac{1}{20T_1}$$

$$\text{Sampling Rate: } \frac{1}{T_1}$$

and then calculate that (Figure 3)

$$\epsilon = 1$$

$$\lambda = 1000$$

$$\log (\lambda / \epsilon) = 6.91$$

$$\left. \begin{aligned} f_r &= \frac{1}{20T_1} \\ f_c &= \frac{1}{25T_1} \end{aligned} \right\} \frac{f_r}{f_c} = \frac{25}{20} = 1.25; \log \left( \frac{f_r}{f_c} \right) = 0.223$$

$$\rightarrow N = \frac{\log (\lambda / \epsilon)}{\log (f_r / f_c)} = 30.97$$

$$\rightarrow N = 31 = \text{Order of the Butterworth Filter}$$

Implementing in cascade, we let  $N = 32$ , which results in 16 sections. Each section will have 4 multipliers so that the overall implementation will require  $16 \times 4 = 64$  multiplies per input sample. For zero phase shift, we can use the input time-reversal procedure. Therefore, the total number of multiply operations will be 128 multiplies per input sample. Given 1500 input samples, we determine that 192 000 multiplies per waveform will be required.

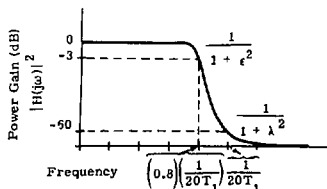


Figure 3  
Recursive (Infinite Impulse Response) Filter



Added to the general criteria is the requirement that the filter operate in minimal time with at least linear, but preferably zero, phase distortion. Of course, the effects of finite word length upon parameter quantization must be considered. Finally, we must have some means for verifying that the system is operating correctly. Since the system will be installed in a computer-production environment, one can expect that either the computer hardware, software, or both will change. Therefore, we require that a system-performance test be available to assure us that a change to the computing environment results in minimal adverse change to the filter/decimator performance. To this end, we propose to examine a test statistic--the ratio of input to output energy for frequencies up to the ultimate Nyquist frequency (i.e., the highest resolvable frequency at the reduced sampling rate).

#### Digital Filter Design

The input sampling rate is  $T_1$ ; we assume that this rate is sufficient to resolve all frequencies in the input waveform (since sufficient information to analyze the original sampling rate is unavailable). The Nyquist frequency for the incoming signal is the  $1/2T_1$ . We want a sample rate reduction of 10 which results in an ultimate Nyquist frequency of  $(1/10)(1/2T_1) = 1/20T_1 = 1/2T_2$  Hz. The low-pass filter must then have a nominal cut-off frequency of  $1/2T_2$  Hz.

There are two types of filters that can be applied: recursive (Infinite Impulse Response, IIR) and nonrecursive (Finite Impulse Response, FIR). Usually, the IIR filter is the faster of the two since it can normally be realized with fewer coefficients and, thereby, fewer multiply operations. However, this design suffers from two difficulties: nonlinear phase distortion and some uncertainty with respect to stability. Both problems can be overcome. The phase distortion can be eliminated by an input time-reversal process in which we run the filter twice but subject the input signal to two time reversals; performing the filter algorithm twice doubles the computing time required. Stability can be assured with careful design and with treatment of parameter quantization effects.

FIR filters are, in general, slower than IIR filters because more coefficients are required; therefore more multiply operations are performed as part of the filter algorithm. However, FIR filters can be designed with linear (easily treatable) phase shifts. Certain procedures can also be implemented to increase the computation speed. The choice between an FIR and IIR filter design then depends upon the function it is required to perform with respect to a given application.

---

\*The actual sampling frequencies are classified and may be found in DR266801, Data Requirements Specification.

### IIR Filter Evaluation

Several types of IIR filters are available, but we use a Butterworth design for its maximally flat passband feature. Based on Stearns' procedure,<sup>5</sup> we established the following specifications.

$$\text{Passband: } 0 - \frac{1}{20T_1}$$

$$\text{Skirt: } \text{Down 3 dB at } (0.8) \left( \frac{1}{20T_1} \right) = \frac{1}{25T_1}$$

$$\text{Stop Band: } \text{Down 60 dB at } \frac{1}{20T_1}$$

$$\text{Sampling Rate: } \frac{1}{T_1}$$

and then calculate that (Figure 3)

$$\epsilon = 1$$

$$\lambda = 1000$$

$$\log (\lambda / \epsilon) = 6.91$$

$$\left. \begin{aligned} f_r &= \frac{1}{20T_1} \\ f_c &= \frac{1}{25T_1} \end{aligned} \right\} \frac{f_r}{f_c} = \frac{25}{20} = 1.25; \log \left( \frac{f_r}{f_c} \right) = 0.223$$

$$\rightarrow N = \frac{\log (\lambda / \epsilon)}{\log (f_r / f_c)} = 30.97$$

$$\rightarrow N = 31 \text{ order of the Butterworth Filter}$$

Implementing this cascade, we let  $N = 32$ , which results in 16 sections. Each section will have 4 multipliers so that the overall implementation will require  $16 \times 4 = 64$  multiplies per input sample. For zero phase shift, we can use the input time-reversal procedure. Therefore, the total number of multiply operations will be 178 multiplies per input sample. Given 1500 input samples, we determine that 192 000 multiplies per waveform will be required.

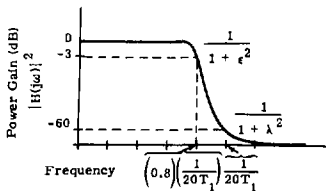


Figure 3  
Recursive (Infinite Impulse Response) Filter

### FIR Filter Evaluation

The FIR designs that are available include the elementary design based on a windowing technique, and an optimal design based on a Remez exchange algorithm.<sup>3</sup> Further, Crochiere and Rabiner<sup>4</sup> have provided a technique for reducing the computing time required for narrow-band filters by implementing the filter/decimator process in a staged or concatenated sequence. This technique results in much less severe filter specifications for each stage than if the entire sample rate reduction were done in a single-step process. As a final option, we can choose between a time- or frequency-domain implementation. In the time domain, we would use a convolution algorithm; in the frequency domain, a Fast Fourier Transform (FFT). In all cases, we use the Remez exchange algorithm for the computation of the filter impulse or unit-sample response, resulting in the filter algorithm-coefficient array.

In general, we employ the following overall specifications which are compatible with the IIR case:

$$\begin{aligned} \text{Passband Ripple Tolerance, } \delta_p &= -80 \text{ dB} \\ \text{Stop Band Ripple Tolerance, } \delta_s &= -80 \text{ dB} \\ \text{End of Passband, } f_p &= \frac{1}{25T_1} \\ \text{Start of Stop Band, } f_s &= \frac{1}{20T_1} \end{aligned}$$

From Crochiere and Rabiner we can obtain an approximate duration of the unit-sample response for an FIR digital filter. If we wish to use a single-stage filter, the duration estimate is 325 samples. Given our 1500-sample-length input waveform, we shall require approximately 487 500 multiplies if implemented via convolution. Only 243 750 multiplies would be required if we take advantage of a symmetrical unit-sample response by using only half the coefficients in our convolution. With the same specifications, we can determine that only 41 367 multiplies are necessary for the frequency domain implementation by FFT.

In the concatenated (or staged) FIR filter/decimator process, instead of reducing the sampling rate by a factor of M in a single process, we first reduce the sampling rate by a smaller amount (say  $M_1$ ) followed by a second reduction by a factor of  $M_2$  and so on. The advantage of the staged process is that much less severe antialiasing filters are required at each stage, resulting in a shorter duration of the unit-sample response. Furthermore, the number of input samples to the second stage is only a fraction of the number presented to the first stage. With each successive stage, fewer input samples are required. The upshot is a process that requires much less computing time than a single-step process.

The results of the two-stage FIR design (detailed in the following section) clearly favor the multistage FIR filter/decimator, implemented via convolution in the time domain. Table 1 summarizes the evaluation process in terms of the required number of products, and Figure 4 illustrates the specific design chosen. Briefly, the input signal with a sampling frequency of  $T_1$  is

rate is reduced to one-half of that of Stage 1. The output from Stage 1 is filtered to remove the aliasing component by a factor of 2 giving the ultimate 50% reduction in sampling rate. Both of the filter realizations are made as part of the convolution algorithm.

Table 1  
Filter-Block Realization Results

Single Stage	Number of Products
FIR - Convolution	487 500
FIR - Convolution (using symmetry)	243 750
FIR (orthogonal, anti-time reversal)	192 000
FIR - FFT	41 367
Two-stage	
FFT Implementation	46 635
Convolution Implementation	34 500

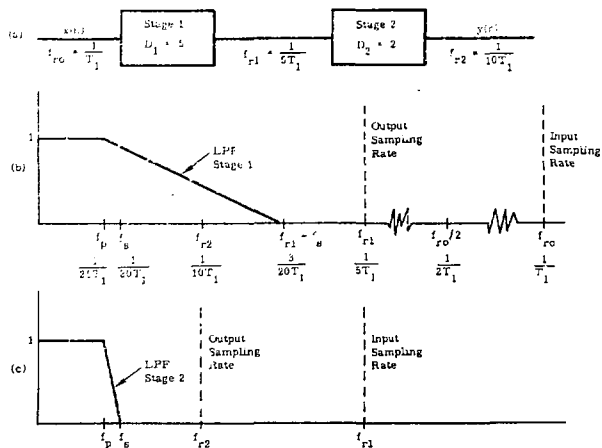


Figure 4. Two-Stage FIR Implementation

### FIR Filter Evaluation

The FIR designs that are available include the elementary design based on a windowing technique, and an optimal design based on a Remez exchange algorithm.<sup>3</sup> Further, Crochiere and Rabiner<sup>4</sup> have provided a technique for reducing the computing time required for narrow-band filters by implementing the filter/decimator process in a staged or concatenated sequence. This technique results in much less severe filter specifications for each stage than if the entire sample-rate reduction were done in a single-step process. As a final option, we can choose between a time- or frequency-domain implementation. In the time domain, we would use a convolution algorithm; in the frequency domain, a Fast Fourier Transform (FFT). In all cases, we use the Remez exchange algorithm for the computation of the filter impulse or unit-sample response, resulting in the filter algorithm-coefficient array.

In general, we employ the following overall specifications which are compatible with the IIR case:

$$\begin{aligned} \text{Passband Ripple Tolerance, } \delta_p &= -60 \text{ dB} \\ \text{Stop Band Ripple Tolerance, } \delta_s &= -60 \text{ dB} \\ \text{End of Passband, } f_p &= \frac{1}{25T_1} \\ \text{Start of Stop Band, } f_b &= \frac{1}{20T_1} \end{aligned}$$

From Crochiere and Rabiner we can obtain an approximate duration of the unit-sample response for an FIR digital filter. If we wish to use a single-stage filter, the ration estimate is 325 samples. Given our 1500-sample-length input waveform, we shall require approximately 427 500 multiplies if implemented via convolution. Only 243 750 multiplies would be required if we take advantage of a symmetrical unit-sample response by using only half the coefficients in our convolution. With the same specifications, we can determine that only 41 387 multiplies are necessary for the frequency domain implementation by FFT.

In the concatenated (or staged) FIR filter/decimator process, instead of reducing the sampling rate by a factor of  $M$  in a single process, we first reduce the sampling rate by a smaller amount (say  $M_1$ ) followed by a second reduction by a factor of  $M_2$  and so on. The advantage of the staged process is that much less severe antialiasing filters are required at each stage, resulting in a shorter duration of the unit-sample response. Furthermore, the number of input samples to the second stage is only a fraction of the number presented to the first stage. With each successive stage, fewer input samples are required. The upshot is a process that requires much less computing time than a single-step process.

The results of the two-stage FIR design (detailed in the following section) clearly favor the multistage FIR filter/decimator, implemented via convolution in the time domain. Table 1 summarizes the evaluation process in terms of the required number of products, and Figure 4 illustrates the specific design chosen. Briefly, the input signal with a sampling frequency of  $T_1$  is

band-limited and decimated by a factor of 5 in Stage 1. The output from Stage 1 is delivered to Stage 2 where decimation by a factor of 2 gives the ultimate 90% reduction in sampling rate. Adjustments for phase delay are made as part of the convolution algorithm.

Table 1  
Filter-Design Evaluation Results

Single Stage	Number of Products
FIR - Convolution	487 500
FIR - Convolution (using symmetry)	243 750
IIR (Butterworth w/time reversal)	192 000
FIR - FFT	41 367

Two-Stage	Number of Products
FFT Implementation	40 635
Convolution Implementation	34 500

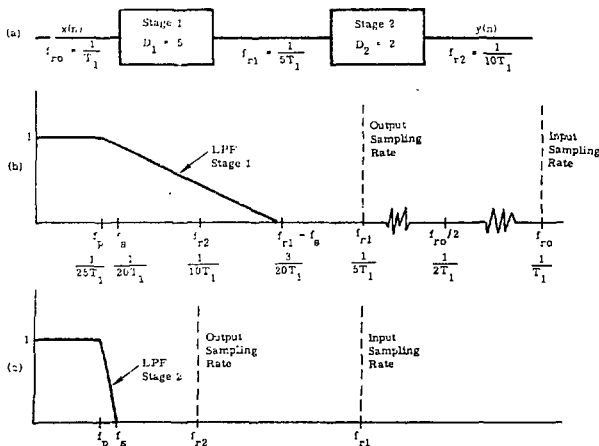


Figure 4. Two-Stage FIR Implementation

Figure 4(a) shows the two processing steps, incorporating both the low-pass filter and the decimation process in each stage. Figure 4(b) shows the power-gain function for Stage 1. The start of the stop band in Stage 1 is set at  $f_{r1} = f_s$  in order to assure that no high frequency components are aliased into the passband. The output sampling frequency at the first stage is  $f_{r1}$ . Figure 4(c) shows the final stage where, as before, the cut-off frequency is set at the ultimate Nyquist frequency. The filter-design program requires normalized frequencies as input. Table 2 gives the relationship between key frequencies and their normalized equivalents. The frequencies are normalized to the relevant input-sampling frequency.

Table 2  
Key Frequencies With Normalized Equivalents

<u>Frequencies</u>	<u>Stage 1</u>	<u>Stage 2</u>
$f_{r0}$	1.0	-
$f_{r0}/2$	0.5	-
$f_{r1}$	0.2	1.0
$f_{r1} - f_s$	0.15	0.75
$f_{r2}$	0.10	0.5
$f_s$	0.05	0.25
$f_p$	0.04	0.20

#### Final Design and Implementation of the FIR Filter

With the decision that a FIR digital filter should be used in this project, we had to determine the particular filter-parameter values that would cause the digital filter to emulate the ideal low-pass filter as nearly as possible. Among these parameters were the filter-ripple specifications and the maximum allowable transition bandwidth. The filter-ripple specifications were established as 0.1% of the frequency response of the filter in both the passbands and stop bands. Currently, the value for the transition bandwidth was set at 20% of the theoretical passband. Both of these values were determined empirically.

The frequency responses of the various filters tested were investigated by using an implementation of the Remez exchange algorithm.<sup>3</sup> Using this software, we were able to generate the filter coefficients after specifying the filter length, the filter type, the frequency limits of the passband and stopbands, the desired frequency responses for the two bands, and the weighting functions for each band. Several examples of the use of this algorithm using various parameter values are summarized in Table 3.

Table 3

Comparison of the Ripple in Both the Passband and Stop Band of  
FIR Digital Low-Pass Filters

Filter Length	Passband			Stop Band		
	Limits	Weighting	Deviation (dB)	Limits	Weighting	Deviation (dB)
327	0 - 0.04	1.0	-59.943	0.05 - 0.5	1.0	-59.943
326	0 - 0.04	1.0	-59.913	0.05 - 0.5	1.0	-59.913
325	0 - 0.04	1.0	-59.914	0.05 - 0.5	1.0	-59.914
71	0 - 0.04	1.0	-18.758	0.05 - 0.5	1.0	-18.758
33	0 - 0.04	1.0	-12.454	0.05 - 0.5	1.0	-12.454
* 33	0 - 0.04	1.0	-64.860	0.15 - 0.5	1.0	-64.860
71	0 - 0.2	64.0	-81.821	0.25 - 0.5	1.0	-45.697
71	0 - 0.2	18.0	-78.185	0.25 - 0.5	1.0	-52.102
71	0 - 0.2	4.0	-88.881	0.25 - 0.5	1.0	-56.840
* 71	0 - 0.2	1.0	-62.890	0.25 - 0.5	1.0	-62.990
71	0 - 0.2	1.0	-57.179	0.25 - 0.5	4.0	-69.220
71	0 - 0.2	1.0	-50.952	0.25 - 0.5	16.0	-75.035
71	0 - 0.2	1.0	-46.652	0.25 - 0.5	64.0	-82.775

\* Filters selected for implementation

This investigation confirmed that a single-stage filter of length 325 (i.e., 325 samples) was required to obtain an attenuation of -60 dB in the stop band. Since the number of computations may often be decreased by increasing the number of filter stages, the multistage design<sup>4</sup> was investigated. The investigation, following the procedure outlined below, indicates the optimal number of stages, the decimation factor associated with each stage, and the length of each stage.

Given:

$$f_{ro} = \text{Original sampling rate} = 1/T_1$$

$$f_{rk} = \text{Ultimate sampling rate} = 0.1/T_1$$

$$\delta_p = \text{Tolerance of magnitude response in the passband} = 0.001$$

$$\delta_s = \text{Tolerance of magnitude response in the stop band} = 0.001$$

$$D = \text{Total decimation factor} = \frac{f_{ro}}{f_{rk}} = 10$$

$$f_s = \text{Lower limit of ultimate stop band} = \frac{f_{rk}}{2} = 0.05/T_1$$

$$f_p = \text{Upper limit of usable passband} = 0.04/T_1$$

$$\Delta f = (f_s - f_p)/f_s = 0.2$$



Procedure:

(A) Calculate  $D_n \left( \frac{\delta_p}{k}, \delta_s \right)$  for

$k$  = number of stages = 1, 2, 3, and where

$$D_n = \left( \delta_1, \delta_2 \right) = \left[ 5.309 \times 10^{-3} \left( \log_{10} \delta_1 \right)^2 + 7.114 \times 10^{-2} \left( \log_{10} \delta_1 \right) - 0.4761 \right]$$

$$\log_{10} \delta_2 = \left[ 2.66 \times 10^{-3} \left( \log_{10} \delta_1 \right)^2 + 0.5941 \left( \log_{10} \delta_1 \right) + 0.4278 \right]$$

$$D \left( \frac{\delta_p}{1}, \delta_s \right) = D(0.001, 0.001) = 3.2558$$

$$D \left( \frac{\delta_p}{2}, \delta_s \right) = L(0.0005, 0.001) = 3.4636$$

$$D \left( \frac{\delta_p}{3}, \delta_s \right) = D(0.00033, 0.001) = 3.5836$$

(B) Determine  $D_i \text{ opt}$ ,  $i = 1, 2, \dots, k$

$$k = 1 \Rightarrow D_1 \text{ opt} = D = 10$$

$$k = 2 \Rightarrow D_1 \text{ opt} = \frac{2D \left( 1 - \sqrt{D\Delta f / (2 - \Delta f)} \right)}{2 - \Delta f (D + 1)} \quad (\text{from Ref. 3})$$

$$= \frac{2 \cdot 10 \left( 1 - \sqrt{10(0.2) / (2 - 0.2)} \right)}{2 - 0.2(10 + 1)}$$

$$= 5.41$$

$$D_2 \text{ opt} = D / D_1 \text{ opt} = 10 / 5.41 = 1.85$$

Since we desire integer declination factors, we chose  $D_1 = 5$  and  $D_2 = 2$ .

$$k = 3 \Rightarrow D_1 = 5, D_2 = 2, D_3 = 1 \quad (\text{from plots given in Ref. 3})$$

(C) Calculate  $R_T$  for  $k = 1, 2, 3$  where

$$R_T = D_n \left( \frac{\delta_p}{k}, \delta_s \right)_{f_{r_0}} \times \sum_{i=1}^k \frac{D_i}{\prod_{j=1}^i D_j \left( 1 - \frac{\delta_s + \epsilon_p}{f_{r_0}} \prod_{j=1}^i D_j \right)}$$

$$\text{(for } k = 1): R_T = 32.558/T_1$$

$$\text{(for } k = 2): R_T = 13.225/T_1$$

$$\text{(for } k = 3): R_T = 17.266/T_1$$

(D) Select  $k$  where  $R_T$  is a minimum

$$k = 2$$

(E) Calculate lengths  $N_i$  of filter stages for selected  $k$  where

$$N_i = \frac{D_n \left( \frac{\delta_p}{k}, \delta_s \right) L_i D_i}{1 - \left[ (2 - \Delta f) / 2D \right] \prod_{j=1}^i D_j} \quad \text{and } L_i = 1, \quad i = 1, 2, \dots, k$$

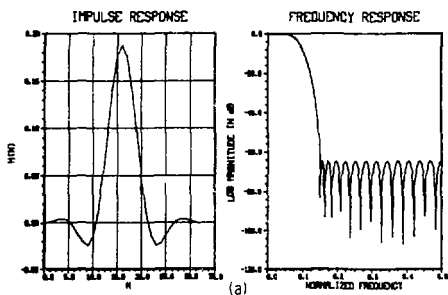
$$N_1 = \frac{3.4636(1)(5)}{1 - \left[ (2 - 0.2) / 2(10) \right] 5} = 31.49$$

$$N_2 = \frac{3.4636(1)(2)}{1 - \left[ (2 - 0.2) / 2(10) \right] 10} = 69.27$$

Thus, from the above calculations it may be concluded that the optimal filter design is a two-stage filter with the decimation factors 5 and 2 associated with the first and second stages respectively. Additionally, we concluded that the lengths of the first and second stages should be approximately 32 and 70, respectively, in order to limit the maximum ripple in the passband and stop band to the specified maximums. Because the phase shift had to be removed, its length had to be a multiple of the sampling interval; therefore, the lengths of the impulse responses for both stages must be odd. As a result, the lengths of the stages selected were 33 and 71.

Using this information, the Remez exchange algorithm was again exercised, resulting in the generation of the impulse and frequency responses depicted in Figure 5. Since the impulse responses have odd lengths, we took advantage of the symmetry and thereby reduced the number of multiplies by a factor of 2; therefore, the number of multiplies required to filter/decimate the 1500-point signal was approximately 35 200. (This value is slightly higher than suggested earlier because the impulse response lengths are now forced to be odd and multiplications are required in the "start-up" of the filter).

LOW-PASS FILTER DESIGN  
 TRANSITION BAND LIMITS - .04000, .15000  
 N-33



LOW-PASS FILTER DESIGN  
 TRANSITION BAND LIMITS - .20000, .25000  
 N-71

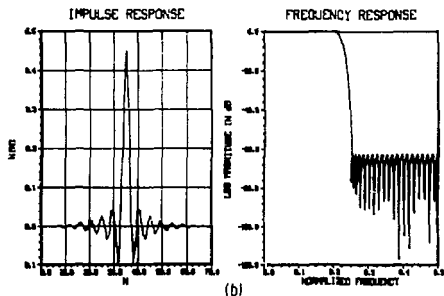


Figure 5. First- and Second-Stage Filter Characteristics

## Experimental Results

After the two-stage digital filter was designed and implemented, initial testing was performed using actual GEND-supplied product-test data. The original signal, the output of the first stage, the output of the second stage, and the associated frequency spectra were plotted. While the results obtained were "reasonable," there was no analytic method by which to test how well the composite filter approximated the ideal one and to assure that the filter coefficients had not been inadvertently changed during a subsequent modification to the software. In order to at least partially fill this void, we developed a statistic that was defined as the ratio of the energy associated with the output signal to the energy of the input signal up to the ultimate Nyquist frequency:

$$E\text{-statistic} = \frac{E_2}{E_1} = 1 - \epsilon,$$

where  $\epsilon$  is a measure of the energy lost during the filtering/decimating process in the frequency domain from zero to one-half the ultimate sampling frequency.

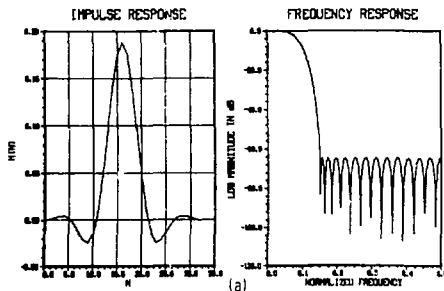
Actual tester-generated data were subjected to the procedure and the values of the E-statistic were calculated, along with their associated mean, sample standard deviation, and the minimum and maximum of the sample set (see results in Figure 6). Examples of the plots of an original signal, the output of the filter, and the corresponding frequency spectra are presented in Figure 7.

Subsequently, we investigated the values of the E-statistic that resulted when signals of known energy content (at least theoretically) were used as input to the system. Generally, test signals should be independent of any disturbances acting upon the system. Since the system is invertible, each signal should produce a unique output for each unique input, and the input signal error should be known. (See Reference 6 for a detailed discussion of test-signal properties.) From the class of available test signals (i.e., impulse or step functions, sinusoids, colored or white noise, etc), a set of white-noise signals seemed the best choice in view of the above test-signal properties. White-noise signals were also preferable because they were unencumbered by the effects of finite-word length or by changes in computing environment (principal disturbances that plagued sinusoidal, impulse, or step signals). The chief concern was that these disturbances might also be coupled to the system.

Thus, we created a set of white-noise signals and subjected them to the filter/decimator system. An example of the resulting time and frequency domain plots, along with the E-statistic summary, are given in Figures 8 and 9.

Using this information, the Remez exchange algorithm was again exercised, resulting in the generation of the impulse and frequency responses depicted in Figure 5. Since the impulse responses have odd lengths, we took advantage of the symmetry and thereby reduced the number of multiplies by a factor of 2; therefore, the number of multiplies required to filter/decimate the 1500-point signal was approximately 39 200. (This value is slightly higher than suggested earlier because the impulse response lengths are now forced to be odd and multiplications are required in the "start-up" of the filter).

LOW-PASS FILTER DESIGN  
 TRANSITION BAND LIMITS - .04000, .15000  
 N-33



LOW-PASS FILTER DESIGN  
 TRANSITION BAND LIMITS - .20000, .25000  
 N-71

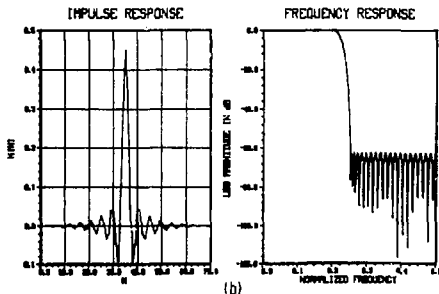


Figure 5. First- and Second-Stage Filter Characteristics

## Experimental Results

After the two-stage digital filter was designed and implemented, initial testing was performed using actual GEND-supplied product-test data. The original signal, the output of the first stage, the output of the second stage, and the associated frequency spectra were plotted. While the results obtained were "reasonable," there was no analytic method by which to test how well the composite filter approximated the ideal one and to assure that the filter coefficients had not been inadvertently changed during a subsequent modification to the software. In order to at least partially fill this void, we developed a statistic that was defined as the ratio of the energy associated with the output signal to the energy of the input signal up to the ultimate Nyquist frequency:

$$\text{E-statistic} = \frac{E_2}{E_1} = 1 - \epsilon,$$

where  $\epsilon$  is a measure of the energy lost during the filtering/decimating process in the frequency domain from zero to one-half the ultimate sampling frequency.

Actual tester-generated data were subjected to the procedure and the values of the E-statistic were calculated, along with their associated mean, sample standard deviation, and the minimum and maximum of the sample set (see results in Figure 6). Examples of the plots of an original signal, the output of the filter, and the corresponding frequency spectra are presented in Figure 7.

Subsequently, we investigated the values of the E-statistic that resulted when signals of known energy content (at least theoretically) were used as input to the system. Generally, test signal would be independent of any disturbances acting upon the system. Since the system is invertible, each signal should produce a unique output for each unique input, and the input signal error should be known. (See Reference 6 for a detailed discussion of test-signal properties.) From the class of available test signals (i.e., impulse or step functions, sinusoids, colored or white noise, etc), a set of white-noise signals seemed the best choice in view of the above test-signal properties. White-noise signals were also preferable because they were unencumbered by the effects of finite-word length or by changes in computing environment (principal disturbances that plagued sinusoidal, impulse, or step signals). The chief concern was that these disturbances might also be coupled to the system.

Thus, we created a set of white-noise signals and subjected them to the filter/decimator system. An example of the resulting time and frequency domain plots, along with the E-statistic summary, are given in Figures 8 and 9.

VARIETYTYPE = 1				
	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1000969830E+01	1001136719E+01	1001103576E+01	6232462266E-04
2	9979013798E+00	1000851336E+01	9999777777E+00	4254974282E-07
VARIETYTYPE = 2				
	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1001097608E+01	1001138415E+01	1001134656E+01	8093635704E-05
2	9997388361E+00	9999916836E+00	99999039352E+00	4651075931E-04

Figure 6. E-Statistic Summary for GEND Product Test Data

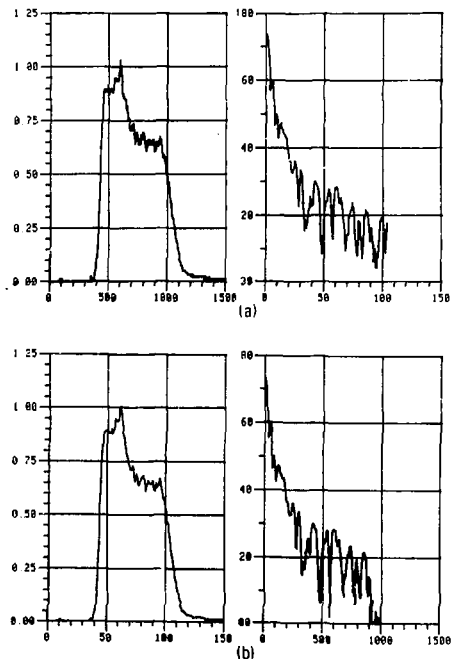


Figure 7. Plots of Original Signal, Output Signal, and Corresponding Frequency Spectra of the Two-Stage Digital Filter

NAUETYPE = 1				
E	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1000909930E+01	.1001136719E+01	.1001103576E+01	.6232462266E-04
2	9979013798E+00	.1000851336E+01	.9999765662E+00	.4254974202E-03
NAUETYPE = 2				
E	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1001097608E+01	.1001138415E+01	.1001134656E+01	.8093635704E-05
2	9997388361E+00	.9999916836E+00	.9999039352E+00	.4651075831E-04

Figure 8. E-Statistic Summary of White-Noise Signals

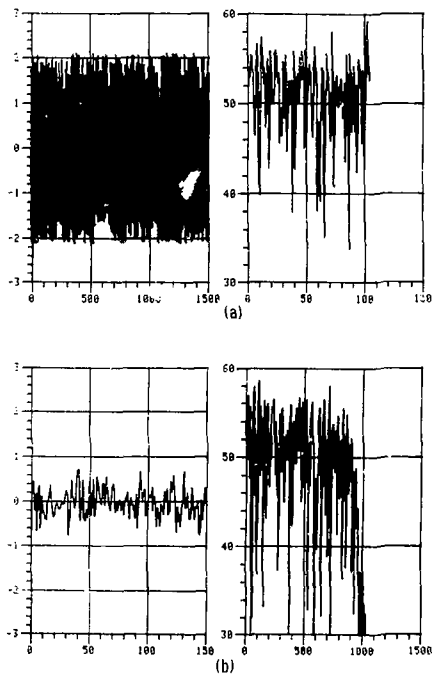


Figure 9. Plots of Original White-Noise Signal, Output Signal, and Corresponding Frequency Spectra of the FIR Filter



NAME TYPE = 1	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1000909930E+01	.1001136719E+01	.1001103576E-01	.6232462266E-04
2	9979013798E+00	.1000851336E+01	.9999765662E-00	4254974282E-07
NAME TYPE = 2	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1001097608E+01	.1001138415E+01	.1001134656E+01	8093635704E-05
2	.9997388361E+00	.9999916836E+00	.9999903935E+00	4651675931E-04

Figure 6. E-Statistic Summary for GEND Product-Test Data

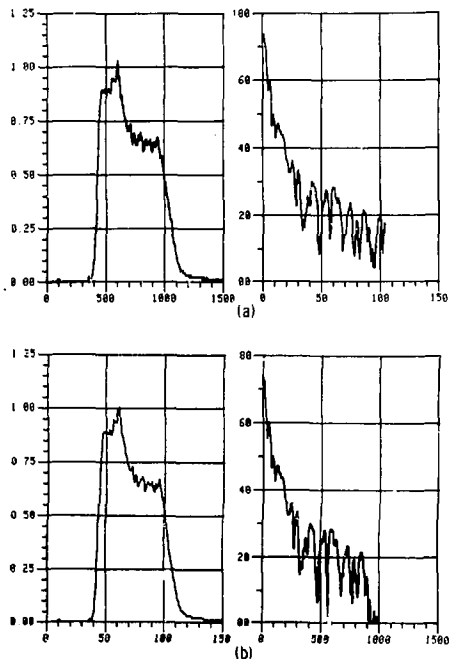


Figure 7. Plots of Original Signal, Output Signal, and Corresponding Frequency Spectra of the Two-Stage Digital Filter

WAVETYPE = 1				
E	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	100099930E+01	1001136719E+01	1001103576E+01	623246226E-04
2	9979013790E+00	1000851336E+01	9999765662E+00	4254974282E-07
WAVETYPE = 2				
E	MINIMUM	MAXIMUM	MEAN	STANDARD DEVIATION
1	1001097600E+01	1001138415E+01	1001134656E+01	8093635704E-05
2	9997322361E+00	9999916836E+00	9999039352E+00	4651075831E-04

Figure 8. E-Statistic Summary of White-Noise Signals

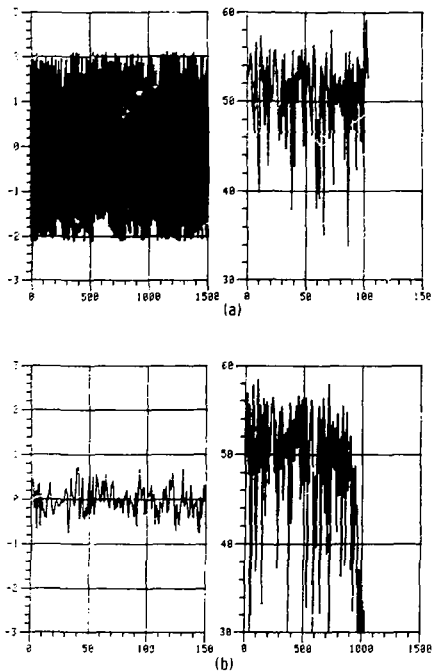


Figure 9. Plots of Original White-Noise Signal, Output Signal, and Corresponding Frequency Spectra of the FIR Filter

The sensitivity of the E-statistic was tested by changing various filter coefficients in one of the filter stages to zero and reproducing the processing summary statistics. The modified filter coefficients and the resulting E-statistic summary are given in Table 4. These results indicate that the sensitivity of the energy statistic to software modifications is highly dependent on the particular filter coefficient modified and the magnitude of the modification. However, we concluded that the energy statistic developed during the course of the project was generally sensitive enough to serve as a qualitative measure of the accuracy of most software modifications that might affect the digital filter routine. In particular, whenever a software modification has been made, the values of the energy statistic resulting from processing the white-noise signals through the newly modified system may be compared to the values of the statistic that were generated by processing the same set of white-noise signals through the filter/decimator prior to the implementation of the modifications. This comparison may take the form of a simple t-test that determines if the differences between corresponding samples from the two sample sets are statistically significant. Additionally, a chi-square test may be used to determine if the two sample sets are from the same distribution.

Table 4  
Effects on the Values of the Energy Statistics Resulting From  
Filter Coefficient Modification

Filter Stage	Modified Coefficient	Original Value	Modified Value	E	Minimum	Maximum	Mean	Standard Deviation
-	-	-	-	1	0.990968	0.998869	0.995796	0.002427
				2	0.776435	0.517711	0.855905	0.146447
1	1	-0.262571E - 03	0.0	1	0.990558	0.998856	0.995763	0.002426
				2	0.776435	0.517674	0.855477	0.146447
1	5	-0.217572E - 01	0.0	1	1.006937	1.029476	1.018613	0.007662
				2	0.807142	0.547400	0.880955	0.047557
1	17	0.188147E - 00	0.0	1	0.651754	0.659314	0.655743	0.001514
				2	0.581716	0.778447	0.669943	0.052294
2	1	-0.245764E - 03	0.0	1	0.990968	0.998869	0.995776	0.002427
				2	0.776437	0.517726	0.858497	0.146451
2	15	-0.933972E - 02	0.0	1	0.950568	0.988869	0.965776	0.002427
				2	0.779782	0.513460	0.850183	0.145418
2	35	0.448985E - 00	0.0	1	0.990868	0.998869	0.995776	0.002427
				2	0.330678	0.439663	0.377498	0.027559

<sup>e</sup> Investigation into the statistical distribution of the energy statistic and consultation with E. E. Ard and R. G. Easterling revealed that it was not a normal distribution, and that, while it might be possible to determine the distribution, it probably would not be an effective use of effort.

### Implementation and Testing of the Filter/Decimator at GEND

After the digital filtering/decimating routine had been implemented and tested on the CDC 6600 under the Network Operating System (NOS), it was transmitted to GEND for implementation on their Honeywell 6000 machine. The version implemented at GEND contained a double-precision filtering algorithm to compensate for the 36-bit word-length machine as opposed to the 60-bit CDC6600; upon implementation, GEND produced values of the energy statistic that were compatible to those obtained at SNLA. Since double-precision computations are expensive relative to single-precision, we used the E-statistic to determine whether or not double-precision computations actually resulted in statistically different outputs. The values generated by a single-precision routine were compared with those produced by the double-precision version via a t-test; the results indicated that the differences in the corresponding values were statistically insignificant. Consequently, the single-precision routine was selected as the production version of the system. This change resulted in an operation time of approximately half the time originally envisioned.

### Conclusions

To provide an antialiasing digital filter for GEND, SNLA personnel developed a two-stage FIR digital filter that was implemented by direct convolution in the time domain and initially tested on SNLA's CDC6600. After this initial testing, which spawned the definition of an "energy statistic," the filter/decimator was implemented on GE's Honeywell 6000 computer, where it was tested and proved for production processing.

As a follow-on project, a system is currently being developed that will provide the means by which the sampling rates of digital signals may be either increased (interpolated) or decreased (decimated).<sup>7</sup> The system, called the Sampling Rate Adjustment of Digital Signals (SRADS), will provide a generalized extension to the software developed in response to the GEND antialiasing filter requirement, and will permit the analysis of groups of digital signals sampled at diverse rates.

#### References

1. J. R. Yoder, Sample Rate Reduction by Decimation: Procedures and Effects, SAND79-0357 (Albuquerque, NM: Sandia Laboratories, 1979).
2. R. W. Shafer and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," Proceedings of the IEEE, Vol. 61, No. 6, June 1973, pp 692-702.
3. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing (New Jersey: Prentice-Hall, 1975).
4. R. E. Crochiere and L. R. Rabiner, "Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow Band Filtering," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol ASSP-23, No. 5, October 1975, pp 444-456.
5. S. D. Stearns, Digital Signal Analysis (Rochelle Park, NJ: Hayden Book Co., 1975).
6. K. J. Aström and P. Eykhoff, "System Identification--A Survey," Automatica, 7, 1971, pp 123-162.
7. B. D. Richard, "Abbreviated Specifications Document for the Sampling Rate Adjustment of Digital Signals System," unpublished phase document, Sandia Laboratories, Albuquerque, NM, November 1979.