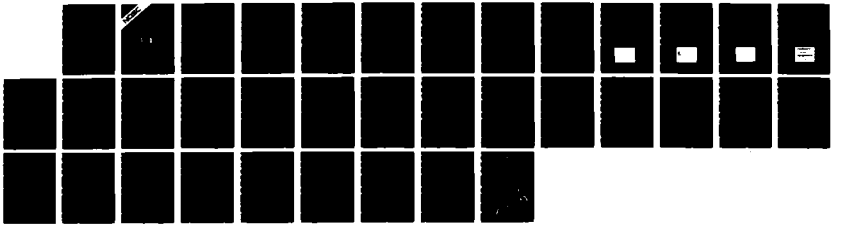
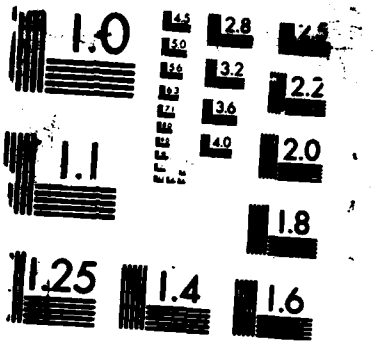
AD-A188 465 WIDE-DYNAMIC-RANGE ANALOG-TO-DIGITAL CONVERSION FOR HFDF(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA W H MCKNIGHT ET AL NOV 86 NOS/TD-1034 1/1

UNCLASSIFIED F/G 17/11 NL





MICROCOPY RESOLUTION TEST CHART

NAVAL OCEAN SYSTEMS CENTER San Diego, California 92152-5000

NOSC TD 1034

Technical Document 1034
November 1986

Wide-Dynamic-Range Analog-to-Digital Conversion for HFDF

W. H. McKnight
D. J. Kaplan
J. M. Speiser
T. O. Jones

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A168 465

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC TD 1034	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Ocean Systems Center 6b OFFICE SYMBOL (if applicable) CODE 743	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000	7b ADDRESS (City, State and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION Space and Naval Warfare Systems Command 8b OFFICE SYMBOL (if applicable) SPWR-6153A	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code) Washington, DC 20363	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 62712N PROJECT NO: XF12151 TASK NO: EE99 AGENCY ACCESSION NO: ICFF 9900

11 TITLE (Include Security Classification)
Wide-Dynamic-Range Analog-to-Digital Conversion for HFDF

12 PERSONAL AUTHOR(S)
W.H. McKnight, D.J. Kaplan, J.M. Speiser, T.O. Jones

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Interim	FROM May 1986 TO Nov 1986	November 1986	37

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	High-frequency direction finder (HFDF), lasers, fiber optics, switches, linear predictive coding, high-speed photoconductors, wide-dynamic-range digitization, noise/jammer cancellation, modems

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Progress is reported in the development of the linear predictive coding/high-speed photoconductor sampling switch concept for wide-dynamic-range digitization of wideband HFDF data, using a synchronously driven laser/fiber-optic system. Development activity reported is in the areas of: (1) simulation, modeling, analysis, and demonstration of noise/jammer cancellation, using linear predictive coding for speech data; (2) assessment of a finite-impulse-response digital filter approach, using a tapped delay line with fixed tap weights and an over-sampled signal to accomplish linear prediction with "arbitrary" accuracy; (3) analysis of the physical properties of photoconductor switches and sample/track-and-hold circuit parameters as they limit the data word accuracy in an A/D converter application; and (4) current and future development activities.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL W.H. McKnight	22b TELEPHONE (Include Area Code) (619) 225-7439
	22c OFFICE SYMBOL Code 743

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

I. INTRODUCTION

This report covers progress in the development of the linear predictive coding/high-speed photoconductor sampling switch concept for wide-dynamic-range digitization of wideband HFDF data, using a synchronously driven laser/fiber-optic system. The background of the concept and previous work is outlined in a report dated 28 March 1983 by McKnight and Speiser [1], as well as in patent disclosures under Navy case numbers 66,297, and 68,315. Development activity reported here is in the areas of: (1) simulation, modeling, analysis, and demonstration of noise/jammer cancellation in speech data by using linear predictive coding; (2) assessment of a finite-impulse-response digital filter approach, using a tapped delay line with fixed tap weights and an over-sampled signal to accomplish linear prediction with "arbitrary" accuracy; (3) analysis of the physical properties of photoconductor switches and sample/track-and-hold circuit parameters as they limit the data word accuracy in an A/D converter application; and (4) current and future development activities.

II. APPLICATION OF A LINEAR PREDICTIVE ALGORITHM FOR REMOVING A JAMMER(S) FROM SPEED DATA

by D.J. Kaplan and W.H. McKnight

A. THEORETICAL BACKGROUND

The linear prediction model is based on the use of an Mth-order linear predictor of the sampled (and digitized) signal s(n), where n denotes the nth sample, which requires a linear combination of the previous M samples (see Markel and Gray [2]). If s(n) denotes the predicted sample and e(n) the prediction error,

$$e(n) = \sum_{i=o}^{M} a_i s(n-i) = s(n) + \sum_{i=1}^{M} a_i s(n-i) = s(n) - \hat{s}(n)$$

where

$$\hat{s}(n) = - \sum_{i=1}^{M} a_i s(n-i)$$

and the coefficients $-a_i$, $i = 1, 2, \ldots, M$ define the predictor coefficients to be found. The minus sign is chosen so that the error is based on a difference of two variables, although this choice is completely arbitrary. The total squared error is given by

$$\alpha = \sum_{n=n_o}^{n_1} e^2(n)$$

where n_o and n_1 define the index limits over which error minimization occurs. We can write

$$\alpha = \sum_{n=n_o}^{n_1} \left[\sum_{i=o}^{M} a_i s(n-i) \right]^2 = \sum_{n=n_o}^{n_1} \sum_{i=o}^{M} \sum_{j=o}^{M} a_i s(n-i) s(n-j) a_j ,$$

and define $c_{ij} = \sum_{n=n_o}^{n_1} s(n-i)s(n-j)$ so that

$$\alpha = \sum_{i=0}^{M} \sum_{j=0}^{M} a_i c_{ij} a_j .$$

Minimization of α is obtained by setting the partial derivative of α with respect to a_k, k=1,2,..., M, to zero and solving the set of M linear simultaneous equations for the unknown predictor coefficients $\{a_i\}$.

$$\frac{\partial \alpha}{\partial a_k} = 2 \sum_{i=0}^{M} a_i c_{ik} = 0, \quad a_o = 1$$

thus $\sum_{i=1}^{M} a_i c_{ik} = -c_{ok}$, k=1,2,..., M. Using the known parameters c_{ik}, i = 0,1,..., M, as defined from the data, one can see that samples s(n) from n_o-M to n_1 are required.

The two specific cases most commonly employed in implementing the general linear predictor model are referred to as the covariance method and the autocorrelation method. Assuming that a sequence of N speech samples $\{s(n)\}$ = s(o), s(1),..., s(N-1) is available, the covariance method is defined by setting n_o = M and n_1 = N-1 so that the error is minimized only over the interval [M, N-1], and all N data samples are used in calculating the covariance matrix elements c_{ij}. In this case, the coefficient set $\{c_{ij}\}$ forms a symmetric semidefinite matrix that will be singular if the input data sequence $\{s(n)\}$ satisfies a linear homogeneous difference equation of order M or less.

The autocorrelation method is defined by setting $n_o = -\infty$ and $n_1 = \infty$ and defining s(n) = o for n < o and n ≥ N. The coefficient set $\{c_{ij}\}$ then forms the elements of a symmetric positive definite Toeplitz matrix, and the coefficients can be expressed in terms of an autocorrelation sequence as

$$c_{ij} = r(|i - j|)$$

re

$$r(k) = \sum_{n=0}^{N-1-k} s(n)s(n+k) \text{ for } k = 0,1,\ldots, N.$$

y values of r(k) for k=0,1,..., M are needed for the solution.

The two methods give identical results when the data sequence is ncated so that s(n) = o for n < M and for n > N - 1 - M. The two methods e similar results when N >> M. Thus the essential differences between se two methods of LPC lie in the manner in which they treat the signal side the analysis interval in the process of determining predictor co-icients. The autocorrelation method requires that the signal be set to o outside the analysis interval and a suitable window, such as a Hamming dow, be used to reduce the abrupt change in signal values occurring at beginning and at the end of the analysis interval. The covariance hod avoids truncations of the signal, but does require that an entire rix of covariances be computed from the data signal. The covariance hod was chosen for this demonstration/simulation/analysis because it oretically gives exact results when the input data consist of exactly n plex sinusoids. The algorithm must be able to handle a situation in ch there are jammers present, but no random signal, so that the input sists of only pure sinusoids.

CODING AND IMPLEMENTATION

Even though the ultimate application of LPC is for the HF band (2-32), where sample rates may be 100 megasamples per second or more, much wledge and insight are to be gained by applying the LPC concept at audio quencies where it is relatively easy to implement. Thus the specific l of this facet of the general development is to develop the appropriate ftware) techniques(s) and to show that a linear predictive algorithm can used to remove a jamming tone (or tones) from speech data. This is some-t similar to a typical HFDF situation in which the jammer may be a strong entional or unintentional tone (or tones), which is (are) coherent over ny" samples, and in which the signal(s) of interest (speech in this case) (are) represented as a low-level signal that is relatively incoherent or erent only over a very few samples. Speech is actually coherent over ically about 20 ms or 200 samples when sampled at 10 kHz (the required imum sampling rate given the speech bandwidth of ~5 kHz). Speech does, ever, offer the advantage of being intelligible so that one can readily e a subjective judgment as to the effectiveness of the jamming tone cellation via LPC. In the case of A/D conversion for HFDF, the jamming e (or tones) would be retained as a predicted quantity (also perhaps luding some of the desired signal), whereas in the present case of speech ner cancellation the predicted jamming signal is subtracted from the oming composite signal and discarded (only the then-unjammed speech is ained).

The finite coherence of speech signals requires a prediction for a signal (speech plus jammer) sample that is far enough into the future so as to well exceed the speech coherence length but not the jammer coherence length. Otherwise, the speech signal would be accurately predicted by the LPC algorithm and canceled (or subtracted), along with the unwanted jammer. Thus we need to employ at least a 200-step predictor. That is, with each new data sample, our linear predictor would produce a data sample prediction of at least 200 steps (samples) in the future.

The procedure for implementing this LPC concept is to first define a block of data from which to calculate the filter weights that best approximate the data. Past data samples are thus weighted to predict a future sample. The order, M, of the filter denotes the number of past samples used to predict a future sample, and the filter weights are determined by minimizing the error between actual data values and predicted data values over a block of N data samples. Of course, M must be sufficient for the algorithm to predict "several" simultaneous jammers. If the block of N samples is not characteristically representative of the data, or if the signal (including noise) statistics change, the filter weights will not produce a very accurate prediction and the filter will need to be reconfigured (a new set of weights calculated). Obviously, it is not likely to be necessary to reconfigure the filter every clock cycle (with each new data point), but the algorithm for doing so must run quickly enough to keep up with the data. One could arrange for the filter weights to be renewed periodically or renewed whenever the prediction error exceeded some predetermined threshold (such as when the dynamic range of an A/D converter is exceeded). In the present case, the filter was reconfigured every clock cycle simply because it was easy and convenient to do so.

The following parameters summarize the choices and operation of this LPC application.

1. The filter order was chosen as M = 2 for the case of a single jamming tone and M = 4 for two jammers. This follows a general rule that requires two filter orders (degrees of freedom) for every complex sinusoid to be canceled.

2. The value of N (the number of past samples used in minimizing the error for determining filter weights) was chosen as 8 for the case of a single jammer (M = 2) and 16 for the case of two jammers (M = 4). This also follows a general recommendation by the authors of the LPC algorithm (Markel and Gray [2]) that the value of N be approximately equal to 4M. Good results were obtained by following this general guideline.

3. Digitized speech data were obtained as a file consisting of 2^{16} (~64k) samples, produced at a rate of 10 kHz (one sample every 0.1 ms). The speech consisted of an adult male voice with a British accent saying "Hello operator, (pause), operator, (pause), hello operator, . . ." Although this is not a standard speech segment, it was felt to be sufficient for this purpose. Signal data were stored in an 8-bit (fixed-point) video "frame store memory" format consisting of 256 lines of data, with each line containing 256 data points. Since we were simply sampling a waveform, all sample values consisted of real numbers (there being no phase or quadrature

information involved). The original unprocessed speech data are shown in figure 1(b). The pixel values range from -128 to +127. Sample values for line #8 are printed in floating-point format in figure 1(a). Figure 1(b) is an image of the entire speech data file (all 256 lines representing ~6.4 seconds of speech), in which the grey level represents the amplitude of the signal. Line #8 occurs near the beginning of the first voiced portion of the speech data file.

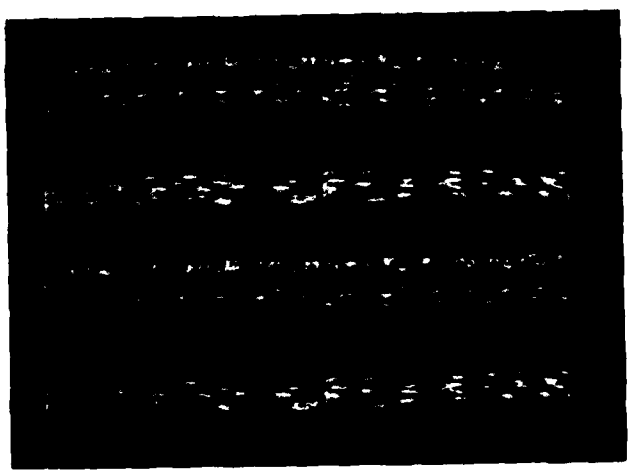
4. One or two digitized (sample rate = 10 kHz) jamming tones were generated in software and stored in a data file. Figure 2 depicts one such pure tone at 1250 Hz, where the amplitude of this jammer was set at 256. Figure 2(a) represents the tone values for line #8 and 2(b) is the video display for this entire data file. The tone amplitude was chosen to give a ratio of jammer to speech amplitude of approximately 2:1, a value that results in the speech being just barely audible (although not intelligible) under the jammer when the two are added together. This sum file is shown in figure 3.

5. Finally, the sum data file represented by figure 3 was processed by an algorithm written by Markel and Gray [2], and the results are shown in figures 4 and 5. Figure 4 shows the predictor error. This algorithm contains two subroutines, AUTO and COVAR. As mentioned previously, COVAR (the so-called covariance method) was chosen for this demonstration/ analysis. The driver algorithm, UNJAM3, and the subroutine, COVAR, are shown in figures 6 and 7, respectively. UNJAM3 uses a second-order (M = 2) filter and minimizes the error over eight samples (N = 8). The filter outputs zeros for the first eight values of the data file, since initially there are insufficient data for an accurate prediction. As can be seen from figure 4(b), the jammer has been well removed. When this file was played out through a D/A converter and recorded on audio cassette tape, the reproduction sounded very close to the original speech, with very little degradation in intelligibility.

Comparing figure 4(a) with figure 1(a), it can be seen that the values of the predictor error are in the same range and have much the same pattern as the original speech sample values. Thus, in this example, the predicted values would not be outside the range of the A/D converter for the low-order bits, as depicted in figures 8 and 9. This indicates that the algorithm would be successful in extending the dynamic range of A/D conversion in this case. The predicted values shown in figure 5, as expected, agree well with the jammer tone values shown in figure 2(a).

-4.0	-8.0	-8.0	-14.0	-16.0	-17.0	-18.0	-16.0	LINE #8
-20.0	-18.0	-17.0	-15.0	-12.0	-13.0	-10.0	-12.0	
-8.0	-6.0	-1.0	1.0	2.0	4.0	3.0	6.0	
6.0	11.0	12.0	13.0	14.0	14.0	15.0	15.0	
15.0	15.0	11.0	9.0	10.0	11.0	12.0	11.0	
10.0	6.0	4.0	4.0	4.0	4.0	2.0	0.0	
0.0	0.0	0.0	0.0	-2.0	-5.0	-6.0	-6.0	
-4.0	-5.0	-5.0	-2.0	0.0	1.0	1.0	0.0	
-2.0	-2.0	-2.0	-2.0	-2.0	-3.0	-4.0	1.0	
-1.0	-1.0	-9.0	-20.0	-26.0	-30.0	-27.0	-25.0	
-29.0	-32.0	-26.0	-20.0	-7.0	-4.0	-5.0	-9.0	
-12.0	-7.0	1.0	5.0	5.0	6.0	8.0	13.0	
16.0	17.0	20.0	19.0	18.0	19.0	17.0	20.0	
21.0	20.0	18.0	12.0	8.0	5.0	2.0	3.0	
3.0	3.0	2.0	1.0	0.0	0.0	1.0	-1.0	
-2.0	-5.0	-5.0	-3.0	-3.0	-1.0	-3.0	-5.0	
-3.0	-1.0	0.0	1.0	2.0	3.0	4.0	6.0	
6.0	5.0	4.0	5.0	4.0	3.0	1.0	-3.0	
-4.0	-2.0	-5.0	-5.0	-25.0	-48.0	-52.0	-55.0	
-36.0	-22.0	-14.0	-12.0	-16.0	-15.0	-4.0	0.0	
4.0	2.0	-4.0	-4.0	-1.0	6.0	13.0	19.0	
17.0	17.0	13.0	14.0	20.0	23.0	22.0	19.0	
11.0	12.0	11.0	13.0	12.0	7.0	1.0	-6.0	
-10.0	-8.0	-2.0	0.0	-1.0	-6.0	-9.0	-8.0	
-7.0	-4.0	-3.0	-5.0	-7.0	-8.0	-5.0	-1.0	
-1.0	-2.0	-1.0	1.0	5.0	8.0	10.0	9.0	
9.0	9.0	10.0	11.0	11.0	9.0	7.0	3.0	
-1.0	-3.0	0.0	1.0	-16.0	-62.0	-89.0	-67.0	
-45.0	-10.0	-1.0	-5.0	-9.0	-19.0	-10.0	6.0	
17.0	13.0	1.0	-15.0	-18.0	-11.0	-2.0	13.0	
25.0	22.0	16.0	12.0	16.0	28.0	32.0	30.0	
26.0	11.0	6.0	3.0	7.0	11.0	8.0	1.0	

(a)

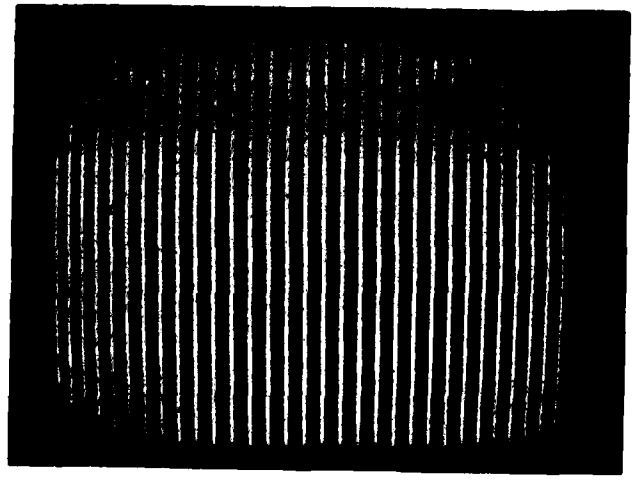


(b)

Figure 1. Original digitized speech data.

0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	LINE #8
.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	f_1 = 1250 Hz
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	AMPLITUDE = 256
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	f_s = 10 kHz
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	
0.0	181.0	256.0	181.0	-0.0	-181.0	-256.0	-181.0	

(a)

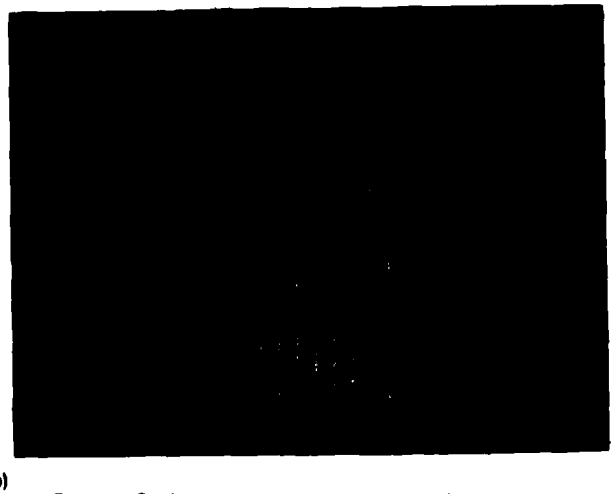


(b)

Figure 2. 1250-Hz jamming tone.

-4.0	173.0	248.0	167.0	-16.0	-198.0	-274.0	-197.0	LINE #8
-20.0	163.0	239.0	166.0	-12.0	-194.0	-266.0	-193.0	
-8.0	175.0	255.0	182.0	2.0	-177.0	-253.0	-175.0	
6.0	192.0	268.0	194.0	14.0	-167.0	-241.0	-166.0	
15.0	196.0	267.0	190.0	10.0	-170.0	-244.0	-170.0	
10.0	187.0	260.0	185.0	4.0	-177.0	-254.0	-181.0	
0.0	181.0	256.0	181.0	-2.0	-186.0	-262.0	-187.0	
-4.0	176.0	251.0	179.0	0.0	-180.0	-255.0	-181.0	
-2.0	179.0	254.0	179.0	-2.0	-184.0	-260.0	-180.0	
-1.0	180.0	247.0	161.0	-26.0	-211.0	-283.0	-206.0	
-29.0	149.0	230.0	161.0	-7.0	-185.0	-261.0	-190.0	
-12.0	174.0	257.0	186.0	5.0	-175.0	-248.0	-168.0	
16.0	198.0	276.0	200.0	18.0	-162.0	-239.0	-161.0	
21.0	201.0	274.0	193.0	8.0	-176.0	-254.0	-178.0	
3.0	184.0	258.0	182.0	-0.0	-181.0	-255.0	-182.0	
-2.0	176.0	251.0	178.0	-3.0	-182.0	-259.0	-186.0	SPEECH + TONE
-3.0	180.0	256.0	182.0	2.0	-178.0	-252.0	-175.0	
6.0	186.0	260.0	186.0	4.0	-178.0	-255.0	-184.0	
-4.0	179.0	251.0	176.0	-25.0	-229.0	-308.0	-236.0	
-36.0	159.0	242.0	169.0	-16.0	-196.0	-260.0	-181.0	
4.0	183.0	252.0	177.0	-1.0	-175.0	-243.0	-162.0	
17.0	198.0	269.0	195.0	20.0	-158.0	-234.0	-162.0	
11.0	193.0	267.0	194.0	12.0	-174.0	-255.0	-187.0	
-10.0	173.0	254.0	181.0	-1.0	-187.0	-265.0	-189.0	
-7.0	177.0	253.0	176.0	-7.0	-189.0	-261.0	-182.0	
-1.0	179.0	255.0	182.0	5.0	-173.0	-246.0	-172.0	
9.0	190.0	266.0	192.0	11.0	-172.0	-249.0	-178.0	
-1.0	178.0	256.0	182.0	-16.0	-243.0	-345.0	-248.0	
-45.0	171.0	255.0	176.0	-9.0	-200.0	-266.0	-175.0	
17.0	194.0	257.0	166.0	-18.0	-192.0	-258.0	-168.0	
25.0	203.0	272.0	193.0	16.0	-153.0	-224.0	-151.0	
26.0	192.0	262.0	184.0	7.0	-170.0	-248.0	-180.0	

(a)



(b)

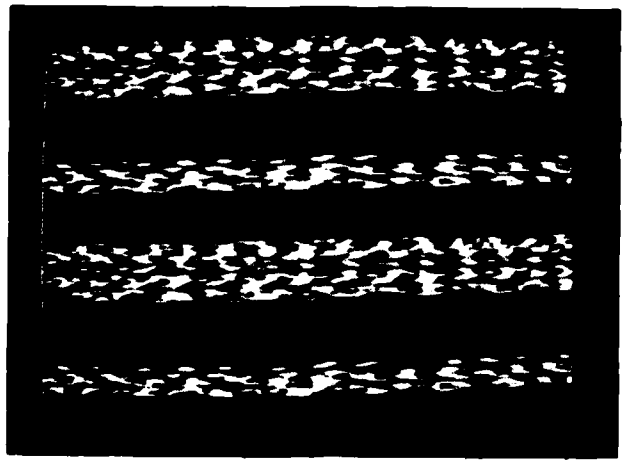
Figure 3. Speech data plus jamming tone.

-4.3	-5.0	-1.1	-9.4	-4.0	-8.5	-9.6	-8.1
-14.4	-6.7	-13.8	-12.6	-12.5	-14.7	-8.1	-13.8
-5.3	-10.3	-4.7	-6.8	-3.6	-0.9	-3.0	3.1
-0.7	6.7	2.0	6.7	7.3	7.1	9.6	9.0
11.2	11.4	8.5	11.4	10.3	8.1	8.8	8.2
9.7	6.2	8.0	6.3	4.6	4.7	3.1	3.7
3.3	0.9	0.9	0.9	-0.6	-0.9	-0.4	-2.2
-1.8	-4.8	-1.8	-1.2	-3.9	-2.7	-1.7	-1.1
-1.5	-0.0	-1.4	-1.0	-0.9	-1.6	-1.4	1.9
-6.3	1.1	-6.6	-5.3	-5.2	-11.8	-10.0	-15.4
-17.2	-14.6	-13.4	-21.5	-14.8	-21.3	-11.9	-11.0
-9.4	-5.3	-6.9	-6.1	-2.3	2.0	1.5	3.6
3.3	7.1	11.4	7.8	11.3	12.8	10.5	17.4
13.5	14.9	14.7	11.3	13.0	9.7	7.9	9.5
5.1	5.4	3.4	3.0	1.8	1.5	1.4	1.2
1.4	-1.8	0.5	-1.3	-3.6	-0.1	-4.2	-1.6
-0.4	-3.2	-2.5	-0.9	-0.1	-0.0	0.3	2.0
1.1	2.6	3.3	4.3	1.7	3.4	2.5	1.0
2.8	1.2	-4.1	1.1	-16.4	-9.6	-6.6	-25.4
-11.2	-27.1	-21.1	-20.9	-23.7	-17.1	-12.5	-16.9
-5.2	-6.1	-4.0	0.4	-2.8	0.8	1.4	5.2
3.1	10.5	5.6	11.4	11.6	9.2	12.4	14.7
11.8	18.2	8.5	12.7	8.9	8.3	8.3	4.0
3.6	2.6	0.8	-3.3	-1.5	-3.3	-2.0	-3.3
-6.0	-2.9	-4.5	-4.6	-3.8	-4.6	-3.0	-4.3
-5.3	-2.0	-0.6	-1.7	-0.2	-0.1	2.5	2.5
5.6	4.9	6.0	6.2	6.6	6.5	7.4	5.0
4.9	4.4	5.0	0.1	-10.3	-21.2	-3.9	-6.4
-39.9	-17.4	-33.5	-19.1	-17.3	-27.0	-8.3	-10.2
-6.4	-4.9	1.4	-1.4	2.0	-3.1	-4.6	4.1
2.7	-1.2	7.1	7.9	10.7	12.9	8.8	17.3
21.2	11.9	20.2	9.9	13.1	9.0	5.8	7.4

LINE #8

PREDICTOR ERROR

(a)



"HELLO OPERATOR,

OPERATOR,

HELLO OPERATOR,

OPERATOR . . . "

(b)

Figure 4. Speech data with jamming tone removed (predicted error).

0.3	178.0	249.1	176.4	-12.0	-189.6	-264.4	-188.9	LINE #8
-5.6	169.7	252.8	178.6	0.5	-179.3	-257.9	-179.3	
-2.7	185.3	259.7	188.9	5.6	-176.1	-250.0	-178.1	
6.7	185.3	266.0	187.4	6.7	-174.1	-250.6	-175.0	
3.8	184.7	258.5	178.6	-0.3	-178.1	-252.8	-178.2	
0.3	180.9	252.0	178.7	-0.6	-181.7	-257.1	-184.7	
-3.3	180.1	255.1	180.1	-1.4	-185.2	-261.6	-184.8	
-2.2	180.9	252.8	180.2	3.9	-177.3	-253.3	-179.9	
-0.5	179.0	255.4	180.0	-1.1	-182.4	-258.6	-181.9	
5.3	178.9	253.6	166.4	-20.8	-199.2	-273.0	-190.6	
-11.8	163.6	243.4	182.5	7.8	-163.7	-249.1	-179.0	
-2.6	179.3	263.9	192.1	7.3	-177.0	-249.5	-171.6	
12.7	190.9	264.6	192.2	6.7	-174.8	-249.5	-178.5	
7.5	186.1	259.3	181.8	-5.0	-185.7	-261.9	-187.5	
-2.1	178.6	254.6	179.0	-1.8	-182.5	-256.4	-180.8	
-3.4	177.8	250.5	179.3	0.6	-181.9	-254.8	-184.4	PREDICTED VALUE
-2.6	183.2	258.5	182.9	2.1	-178.0	-252.3	-177.1	
4.9	183.4	256.7	181.7	2.3	-181.4	-257.5	-185.0	
-6.8	177.8	255.1	174.9	-8.6	-219.4	-301.4	-210.6	
-24.8	186.1	263.1	189.9	7.7	-178.9	-247.5	-164.1	
9.2	189.1	256.0	176.6	1.8	-175.8	-244.4	-167.2	
13.9	187.5	263.4	183.6	8.4	-167.2	-246.4	-176.7	
-0.8	174.8	258.5	181.4	3.1	-182.3	-263.3	-191.1	
-13.6	170.4	253.2	184.3	0.5	-183.7	-263.0	-185.7	
-1.0	180.0	257.5	180.7	-3.2	-184.5	-258.0	-177.8	
4.3	181.0	255.6	183.7	5.2	-172.9	-248.5	-174.5	
3.4	185.2	260.0	185.8	4.4	-178.5	-256.4	-183.0	
-5.9	173.6	251.0	182.0	-5.7	-221.8	-341.1	-241.7	
-5.1	188.4	288.5	195.1	8.3	-173.0	-257.7	-164.9	
23.4	198.9	255.6	167.5	-20.0	-188.9	-253.4	-172.1	
22.3	204.3	264.9	185.1	5.2	-165.9	-232.8	-168.3	
4.8	180.1	241.8	174.1	-6.1	-179.0	-253.8	-187.4	

Figure 5. Predicted data (jamming tone).

```
C       This program UNJAMs or removes an interfering
C       jamming tone from a digitized speech file.
        PROGRAM UNJAM3
        DIMENSION X(8),A1(256),A(3),GRC(2),E(256),P(256)
        OPEN(UNIT=10,ACCESS='DIRECT',RECORDSIZE=256,
     1  MAXREC=256,TYPE='OLD')
        OPEN(UNIT=11,ACCESS='DIRECT',RECORDSIZE=256,
     1  MAXREC=256,TYPE='NEW')
        OPEN(UNIT=12,ACCESS='DIRECT',RECORDSIZE=256,
     1  MAXREC=256,TYPE='NEW')
        M=2
        N=8
C       output zeroes in the first 8 positions
        DO 1 J=1,8
        E(J)=0.0
        P(J)=0.0
1       CONTINUE
C       compile the first 8 sample points
        READ(10'1)A1
        DO 2 J=1,8
        X(J)=A1(J)
2       CONTINUE
C       do the first line
        I=1
C       update the sample vector
        DO 4 J=9,256
        DO 3 K=1,7
        X(K)=X(K+1)
3       CONTINUE
        X(8)=A1(J)
C       calculate the filter coefficients
        CALL COVAR(N,X,M,A,ALPHA,GRC)
C       predict the next sample and calculate
C       the predictor error.
C       PRINT 10,J
C       PRINT 9,A(2),A(3)
C       PRINT 9,X(7),X(6)
        P(J)=-A(2)*X(7)-A(3)*X(6)
        E(J)=X(8)-P(J)
C       PP=P(J)
C       EE=E(J)
C       TYPE 10,J
C10     FORMAT(1X,'J=',I12)
C       PRINT 9,PP,EE
C9      FORMAT(1X,2E16.6)
4       CONTINUE
        WRITE(11'I)P
        WRITE(12'I)E
        TYPE 5,I
5       FORMAT('+',I12)
C       GO TO 6
C       begin main loop
        DO 6 I=2,256
        READ(10'I)A1
        DO 7 J=1,256
        DO 8 K=1,7
        X(K)=X(K+1)
8       CONTINUE
        X(8)=A1(J)
C       calculate the filter coefficients
        CALL COVAR(N,X,M,A,ALPHA,GRC)
C       predict the next sample and calculate
C       the predictor error.
        P(J)=-A(2)*X(7)-A(3)*X(6)
        E(J)=X(8)-P(J)
7       CONTINUE
        WRITE(11'I)P
        WRITE(12'I)E
        TYPE 5,I
6       CONTINUE
        END
```

Figure 6. Driver program.

```
C-----------------------------------------------------------------------
C SUBROUTINE:  COVAR
C A SUBROUTINE FOR IMPLEMENTING THE COVARIANCE
C METHOD OF LINEAR PREDICTION ANALYSIS
C-----------------------------------------------------------------------
C
      SUBROUTINE COVAR(N, X, M, A, ALPHA, GRC)
C
C   INPUTS:   N - NO. OF DATA POINTS
C             X(N) - INPUT DATA SEQUENCE
C             M - ORDER OF FILTER (M<21, SEE NOTE*)
C  OUTPUTS:   A - FILTER COEFFICIENTS
C             ALPHA - RESIDUAL "ENERGY"
C             A - FILTER COEFFICIENTS
C             GRC - "GENERALIZED REFLECTION COEFFICIENTS",
C
C *PROGRAM LIMITED TO M=20, BECAUSE OF THE DIMENSIONS
C B(M*(M+1)/2), BETA(M), AND CC(M+1)
C
      DIMENSION X(1), A(1), GRC(1)
      DIMENSION B(190), BETA(20), CC(21)
      MP = M + 1
      MT = MP*M/2
      MT = (MP*M)/2
      DO 10 J=1,MT
        B(J) = 0.
   10 CONTINUE
      ALPHA = 0.
      CC(1) = 0.
      CC(2) = 0.
      DO 20 NP=MP,N
        NP1 = NP - 1
        ALPHA = ALPHA + X(NP)*X(NP)
        CC(1) = CC(1) + X(NP)*X(NP1)
        CC(2) = CC(2) + X(NP1)*X(NP1)
   20 CONTINUE
      B(1) = 1.
      BETA(1) = CC(2)
      GRC(1) = -CC(1)/CC(2)
      A(1) = 1.
      A(2) = GRC(1)
      ALPHA = ALPHA + GRC(1)*CC(1)
      MF = M
      DO 130 MINC=2,MF
        DO 30 J=1,MINC
          JP = MINC + 2 - J
          N1 = MP + 1 - JP
          N2 = N + 1 - MINC
          N3 = N + 2 - JP
          N4 = MP - MINC
          CC(JP) = CC(JP-1) + X(N4)*X(N1) - X(N2)*X(N3)
   30   CONTINUE
        CC(1) = 0.
        DO 40 NP=MP,N
          N1 = NP - MINC
          CC(1) = CC(1) + X(N1)*X(NP)
   40   CONTINUE
        MSUB = (MINC*MINC-MINC)/2
        MM1 = MINC - 1
        N1 = MSUB + MINC
        B(N1) = 1.
        DO 80 IP=1,MM1
          ISUB = (IP*IP-IP)/2
          IF (BETA(IP)) 150, 150, 50
   50     GAM = 0.
          DO 60 J=1,IP
            N1 = ISUB + J
            GAM = GAM + CC(J+1)*B(N1)
   60     CONTINUE
          GAM = GAM/BETA(IP)
          DO 70 JP=1,IP
            N1 = MSUB + JP
            N2 = ISUB + JP
            B(N1) = B(N1) - GAM*B(N2)
   70     CONTINUE
   80   CONTINUE
        BETA(MINC) = 0.
        DO 90 J=1,MINC
          N1 = MSUB + J
          BETA(MINC) = BETA(MINC) + CC(J+1)*B(N1)
   90   CONTINUE
        IF (BETA(MINC)) 150, 150, 100
  100   S = 0.
        DO 110 IP=1,MINC
          S = S + CC(IP)*A(IP)
  110   CONTINUE
        GRC(MINC) = -S/BETA(MINC)
        DO 120 IP=2,MINC
          M2 = MSUB + IP - 1
          A(IP) = A(IP) + GRC(MINC)*B(M2)
  120   CONTINUE
        A(MINC+1) = GRC(MINC)
        S = GRC(MINC)*GRC(MINC)*BETA(MINC)
        ALPHA = ALPHA - S
        IF (ALPHA) 150, 150, 130
  130 CONTINUE
  140 RETURN
  150 CONTINUE
C
C WARNING - SINGULAR MATRIX
C
C     PRINT 9999
C9999 FORMAT (34H WARNING - SINGULAR MATRIX - COVAR)
      GO TO 140
        END
```

Figure 7. Covariance method subroutine.

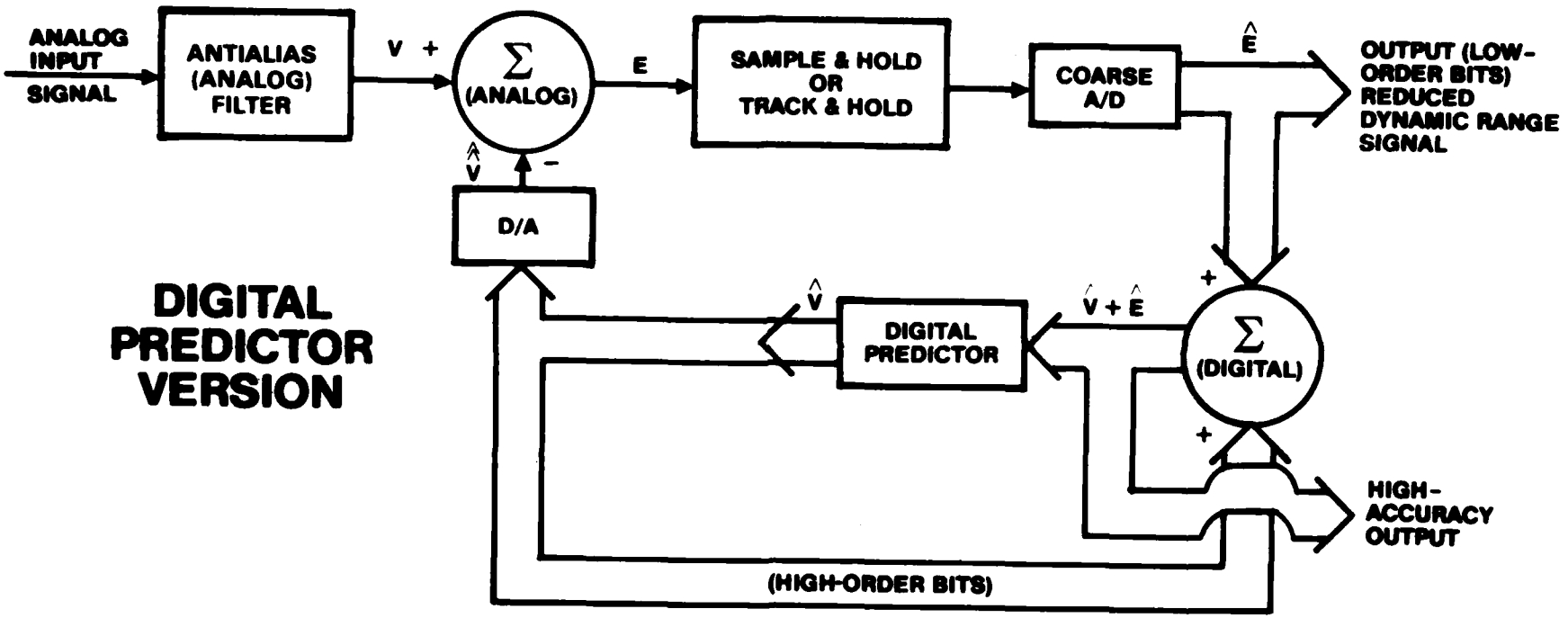


Figure 8. Closed-loop or "Feedback" configuration for linear predictive analog-to-digital converter (1).

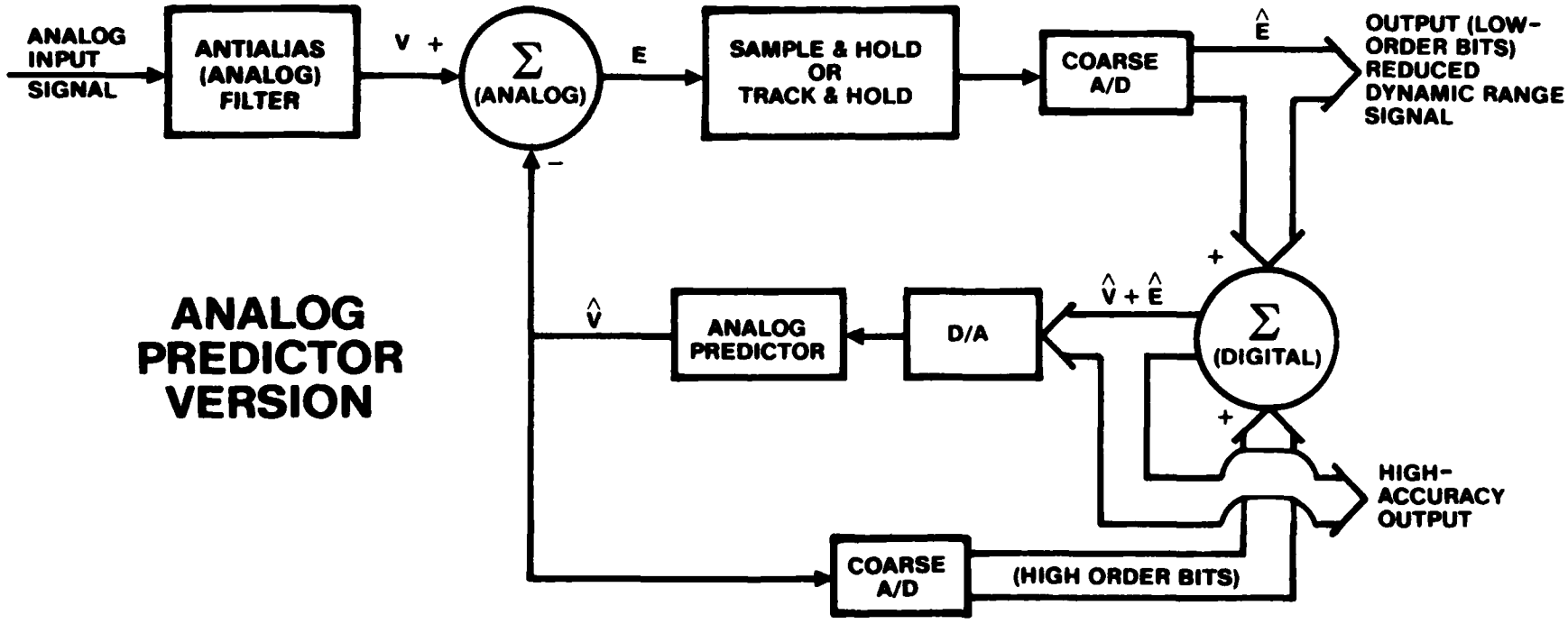


Figure 9. Closed-loop or "Feedback" configuration for linear predictive analog-to-digital converter (2).

C. RESULTS AND CONCLUSIONS

During this LPC voice implementation study/analysis, the following parameters were varied. In all cases, the jamming spectrum was constant during the entire data file.

1. M - the order of the filter (the number of past data samples used in the prediction calculation).

2. N - the number of data samples over which the error is minimized.

3. The number of jamming tones—1 or 2.

4. The frequency of the jammer(s).

5. The amplitude of the jammer(s) relative to that of the speech.

6. P - the number of steps ahead predicted by the algorithm (single step or multistep).

7. The frequency with which the filter coefficients are recalculated.

Results related to each of the above factors were as follows:

1. As is known theoretically, M must be at least twice the number of complex sinusoids in the input data. This was observed experimentally. When there were two jamming tones (one at 1250 Hz and one at 1406 Hz) added to the speech data and M was set at 2, the algorithm was unable to remove both tones effectively and a buzzing tone was heard throughout the file corresponding to the beat frequency between the two jammers. With M = 4, the buzzing was eliminated but the intelligibility of the speech was less than for the M = 2 case. This result occurs because speech can be well approximated as a sum of M' pure sinusoids, where M' is typically on the order of 10–20. Thus, as M is increased, the LPC algorithm becomes a better predictor, not only of the jammer(s) but also of the speech as well. The predictor error therefore becomes smaller. This is a consequence of speech not being well modeled in this case as a random residual since it has a finite coherence length. However, for the HFDF application, the residual should be well modeled as a more truly random variable, and, in that case, the order of the filter should be large enough to handle the number of degrees of freedom that the jammers represent.

2. As stated by Markel and Gray [2], N must be at least four times greater than M for good results. This was observed to be true in our results. However, N must be kept as small as possible, consistent with this requirement, to keep calculation times to a reasonable level. It was found that it took 8.5 minutes to run the LPC program on a 64-k sample data file with M = 2 and N = 8 (single jammer) when the filter coefficients were recalculated every clock cycle. If one does not reconfigure the filter every clock cycle (and there is no need to if the jammer spectrum remains reasonably constant), one can increase both M and N with-

out adversely affecting the processing time. Theoretically, M should be made large enough that M samples will represent a time span that is large compared to the coherence time of the residual.

3. The effect of increasing the number of jamming tones was to require increasing M, as explained above. If the jammers are bands rather than lines, additional requirements will be placed on the LPC algorithm. This case has not yet been investigated.

4. Changing the frequency of the jammer had some effect on the intelligibility of the unjammed speech. When the jammer frequency, f, was set to 664 Hz, the intelligibility of the speech was degraded because f was close to the fundamental frequency of the speech (~500 Hz). This occurs mainly because a significant part of the speech spectrum is falling into the passband of the filter and hence is being subtracted along with the jammer. Again, changing the frequency of the jammer should not present a problem for truly random or broadband signals.

5. Varying the amplitude of the jammer(s) relative to that of the speech did not seem to have much effect on the intelligibility of the unjammed speech as long as that amplitude was no lower than about 2:1. This was to be expected because the LPC algorithm works on the power in the coherent part of the signal. Since the speech is also highly coherent over the filter length, it is not surprising that the LPC algorithm would break down (partially predict the speech signal and subtract it along with the jammer) for low-amplitude jammers. Jammer-to-speech amplitude ratios as large as 256:1 were tried and, with proper values of M and N, it was possible to accurately reproduce the speech to an intelligible degree.

6. P was varied in an attempt to overcome the problems associated with the coherence of the speech. Values used were P = 1, 32, and 2000. At P = 2000, the speech at 2000 samples ahead should have been completely decorrelated from its present value. However, not much improvement was noticed in the intelligibility of the speech and, in fact, an additional problem was observed since an echo of the jammer was produced at the end boundaries of speech segments. Thus multistep prediction, at least for large values of P, does not seem to be appropriate for this application. It would be even less appropriate for the HFDF application, in which the coherence time of expected residuals is very short and the high sample rate will not allow the slow algorithms required for multistep prediction.

7. The times at which the filter is to be reconfigured were found to be very important in the problem of removing a jammer from speech. If this type of LPC technique were to be used for HFDF signals, it would also be important. If the filter is reconfigured during a period when the signal statistics are nonstationary, as during a boundary between speech and silent segments, the filter will be reconfigured incorrectly and the prediction error will exceed the

dynamic range of the residual A/D converter. This was born out by observation in this situation.

In conclusion, the following observations become apparent for application of this LPC technique to HFDF data:

1. The filter order (M) and the minimization length (N samples) should be kept as short as possible to accomplish the objectives of digitizing the total signal to the required accuracy.

2. Single-step prediction should be used (if possible) for greater prediction accuracy.

3. The filter weights should not be recalculated every clock cycle but, rather, only when the residual exceeds the range of the corresponding A/D converter.

III. REAL-TIME SIGNAL PREDICTION FOR HF A/D CONVERSION

by J.M. Speiser

A. LINEAR PREDICTION TECHNOLOGY

We have previously addressed the use of linear prediction to increase the dynamic range and precision of analog-to-digital converters [1]. It was shown that the dynamic range or precision could be improved (in units of power) by a factor equal to the ratio of the process variance to its one-sample prediction error variance.

Two principal concerns in applying this technique are (a) ideal predictor performance, and (b) real-time implementation of the predictor when the sampling rate is high.

For a wide-sense stationary random process with spectral density function s(f), the dynamic range improvement or ratio of input random process variance to prediction error variance is the ratio of the arithmetic mean of s(f) to its geometric mean [3].

Initial candidates for the predictor realization were as follows.

1. Recursively updating an estimate of the sample covariance function and solving the normal equations by using Durbin's algorithm [4]. This would require on the order of N-squared multiply-adds per prediction when N past samples are used—not feasible for high-speed real-time implementation.

2. Nonreal-time estimation of the covariance function and solution of the equation for prediction coefficients. This requires the assumption of stationarity over an extended period, and could therefore result in unsatisfactory tracking if the stationarity period was less than the sum of the covariance update time and the prediction equation solution time.

3. Use of an adaptive lattice filter (preferably a least-squares lattice rather than a gradient descent lattice filter) to provide order of N parallelism in solving the prediction problem [5]. However, the least-squares lattice is a relatively complicated structure, and requires additional computations to update the reflection coefficients.

In summary, the previously proposed predictors have two areas of difficulty: (a) complexity and consequent difficulty of providing a real-time implementation with a latency of less than the sampling interval, when the sampling rate is high; and (b) difficulty of guaranteeing prediction performance without prior knowledge of the spectrum of the input process.

It appears that both of these difficulties can be circumvented by using some little-known results in prediction theory attributable to Brown [6] and Splettstosser [7].

Brown showed that any deterministic signal or wide-sense stationary random process sampled at a rate above the Nyquist rate can be approximated arbitrarily well by a linear combination of its past samples (see his ref 5). In fact, he provides two choices of such weights that may be computed without knowing the signal spectrum. Only a knowledge of the oversampling ratio is required. One set of weights is given explicitly for the case when the sampling is at a rate exceeding twice Nyquist:

$$x_n = \sum_{k=1}^{n} a_{kn}\, x(t-kT)$$

where

$$a_{kn} = (-1)^{k+1}\,(\cos \pi T)[n!/k!(n-k)!],$$

T is the normalized sampling interval, and n is the number of past samples used in the prediction. For this method, it is required that $T < 1/2$.

Brown gives a second method for choosing the prediction coefficients for any sampling rate above Nyquist (i.e., $T < 1$) as the solution of a set of linear equations requiring only knowledge of the sampling interval and number of past samples to be used in the predictor:

$$\sum_{k=1}^{n} a_{kn}\, q(k-p) = q(p) \qquad \text{for } p = 1,2,\ldots,n$$

where

$$q(p) = \sin(\pi pT)/(\pi pT).$$

Brown shows that for either of the above sets of prediction coefficients, the transfer function of the prediction error filter converges uniformly (in frequency) to zero as the number of past samples, n, becomes large. This guarantees that the mean-squared prediction error converges to zero. For the second set of predictor coefficients, he also shows that the

integral of the magnitude squared of the prediction error filter transfer function is minimized for each choice of the number of past samples utilized, thus minimizing the Cauchy-Schwarz bound for the mean-squared prediction error.

Splettstosser exhibits a closed-form solution for asymptotically good predictor weights for any sampling rate above 1.5 Nyquist (i.e., $T < 2/3$), but does not show a desirable extremal property for his weights for each value of n, the number of past samples used.

The reason that Brown and Splettstosser are able to obtain perfect prediction via oversampling is that the spectrum of an oversampled signal has a region of nonzero extent where it is zero. The prediction error results in Hannan's book [2] then tell us that perfect prediction is possible. More specifically, when the (un-normalized) sampling interval T is less than $1/(2W)$, the spectrum is zero between W and the folding frequency $1/(2T)$. Since the signal spectrum is lowpass with cutoff frequency W, and if the prediction error filter is chosen to be a highpass filter with cutoff frequency between W and the folding frequency $1/(2T)$, the output of the prediction filter has zero error. The closer $1/(2T)$ is to W, the sharper the cutoff characteristic required for the prediction error filter—and hence the larger the number of taps that will be required.

The above discussion shows that when the signal is oversampled, there is a great deal of flexibility in the choice of prediction filter with small prediction error. For application to the predictive A/D converter, it is desirable that the predictor be robust with respect to perturbation in its input data, since the prediction process will have to start up by using only coarsely quantized past samples. In view of the difficulty of analyzing a feedback loop with imbedded nonlinearity, it will be necessary to simulate different choices of prediction filter coefficients.

Since it is crucial for the predictive A/D application to keep the latency less than the sampling interval, a strong candidate implementation is a hybrid digital/analog transversal filter employing digital multiplication but analog summation. While digital summation is preferable with respect to accuracy, a binary tree of adders to sum n terms has a latency of $\log_2 n$ addition times. Even though the throughput is adequate, the latency is inadequate when the sampling period is comparable to an arithmetic operation time.

B. MATHEMATICS

1. Hannan's prediction error summary used the spectrum specified in terms of radian frequency. It is both simpler and more convenient to use those results converted to ordinary (normalized) frequency.

Let the correlation function for a random sequence be

$$r(t) = \int_{-1/2}^{1/2} e^{i2\pi ft}\, s(f)\, df$$

where s(f) is the spectral density function of frequency f.

estimation situation	variance of prediction error	
no knowledge	$\int_{-1/2}^{1/2} s(f)\, df$	[arithmetic mean of s(f)]
knowledge of all past	$\exp\left[\int_{-1/2}^{1/2} \log[s(f)]df\right]$	[geometric mean of s(f)]
knowledge of past and future	$\frac{1}{\int_{-1/2}^{1/2} \frac{1}{s(f)}df}$	[harmonic mean of s(f)]

2. Oversampling (deterministic case):

$$\text{Let } h(t) = \sum_{m=-\infty}^{\infty} \delta(t-mT)$$

$$H(f) = \int_{-\infty}^{\infty} h(t)\, e^{-i2\pi ft}\, dt = (1/T) \sum_{n=-\infty}^{\infty} \delta[f-(n/T)]$$

(See M.J. Lighthill, Fourier Analysis and Generalized Functions, Cambridge University Press, 1964, pp. 67-68)

$$\text{Let } g(t) = \int_{-\infty}^{\infty} G(f)\, e^{i2\pi ft}\, df$$

$$g_1(t) = \int_{-W}^{W} G(f)\, e^{i2\pi ft}\, df = \int_{-\infty}^{\infty} G_1(f)\, e^{i2\pi ft}\, df$$

where

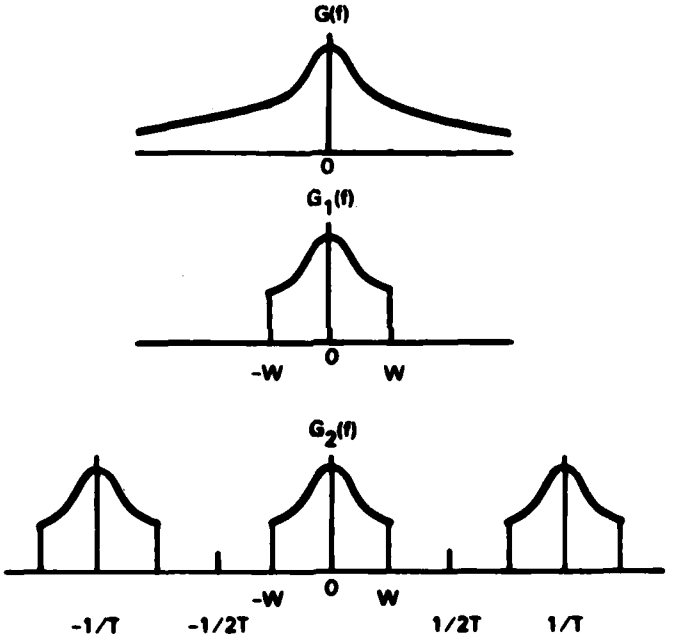
$$G_1(f) = \begin{cases} G(f), & |f| < W \\ 0, & |f| > W \end{cases}$$

and $g_1(t)$ is a lowpass filtered version of $g(t)$.

Let $g_2(t) = g_1(t)\ h(t)$ = sampled version of $g_1(t)$ with sampling interval T.

$$G_2(f) = G_1(f) * H(f) = (1/T) \sum_{n=-\infty}^{\infty} G_1[f-(n/T)].$$

The situation is illustrated pictorially below, assuming $T < (1/2W)$ so that $g_2(t)$ is oversampled.



$G_2(f)$ is periodic with period 1/T. The frequency 1/(2T) is called the folding frequency.

$$G_2(f) = \begin{cases} (1/T)G_1(f), & |f| < W \\ 0 & W < |f| < (1/2T) \end{cases}.$$

Note the following spectral density function for an oversampled, wide-sense stationary (wss) random process. Let x(t) be a wss random process with autocorrelation function r(t) and spectral density function s(f).

$$r(t) = E[x(u)\ x(u+t)] = \int_{-\infty}^{\infty} s(f)\ e^{+i2\pi ft}\ df$$

where E denotes statistical expectation.

Let $x_1(t)$ be a lowpass filtered version of x(t), bandlimited to [-W,W], with corresponding autocorrelation function $r_1(t)$ and spectral density function $s_1(f)$.

$$r_1(t) = E\left[x_1(u)\ x_1(u+t)\right] = \int_{-W}^{W} s(f)\ e^{i2\pi ft}\ df = \int_{-\infty}^{\infty} s_1(f)\ e^{i2\pi ft}\ df$$

$$s_1(f) = \begin{cases} s(f), & |f| < W \\ 0, & |f| > W \end{cases} .$$

Next, consider the random sequence obtained by sampling $x_1(t)$ at multiples of T, where $T < 1/(2W)$.

$$E\left\{x_1(kT)\ x_1[(k+n)T]\right\} = r_1(nT) = \int_{-W}^{W} s(f)e^{i2\pi fnt}\ df .$$

Since $T < 1/(2W)$, it is also true that $W < 1/(2T)$.

Define $$s_2(f) = \begin{cases} s(f), & |f| < W \\ 0, & W < |f| < 1/(2T) \end{cases}$$

$$r_1(nt) = \int_{-1/(2T)}^{1/(2T)} s_2(f)\ e^{i2\pi fnT}\ df .$$

In other words, the spectral density function for the oversampled random process is zero at all frequencies between W and 1/(2T).

The autocorrelation sequence of the sampled process may also be expressed in terms of the normalized spectral density function:

$$r_1(nT) = \int_{-1/2}^{1/2} s_3(u)\ e^{i2\pi nu}\ du$$

$$s_3(u) = \begin{cases} Ts(u/T), & |u| < TW \\ 0, & TW < |u| < 1/2 \end{cases} .$$

IV. ANALYSIS OF InP PHOTOCONDUCTOR SWITCHES FOR SIGNAL SAMPLING

by T.O. Jones and W.H. McKnight

This analysis is intended to offer a first consideration of signal sample accuracy (dynamic range) limitations arising from the physical properties of a photoconductor sample-/track-and-hold switch and the other circuit elements in the sample-/track-and-hold circuit. For the purpose of this analysis, we will assume that a typical sample-/track-and-hold circuit consists of four basic elements: an input preamplifier, the photoconductor switch, a hold capacitor, and an output amplifier. Figure 10 illustrates the corresponding circuit schematic where

R_0 ≡ output impedence of analog preamplifier

$V(t)$ ≡ signal source to be sampled (e.g., V_o sin wt)

$R_S(on)$ ≡ resistance (steady state) of photoconductor switch when illuminated (switch closed)

$R_S(off)$ ≡ resistance (steady state) of photoconductor switch attributable to thermally generated carriers (unilluminated or open-switch state)

C_S ≡ capacitance of photocondutor switch

C_H ≡ capacitance of hold capacitor

R_A ≡ input resistance of sample output amplifier.

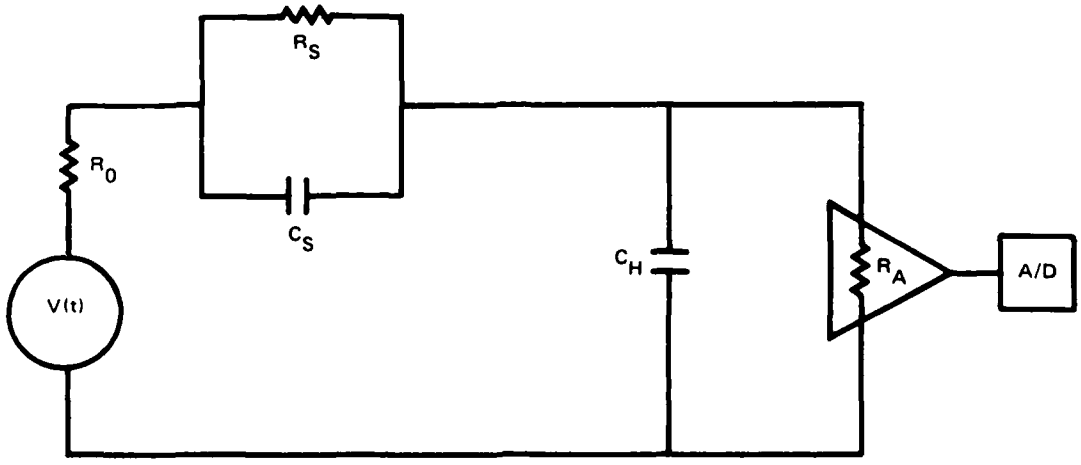


Figure 10. Sample and hold schematic circuit.

This circuit functions under a square-wave clock (e.g., mode-locked ion argon or Nd YAG laser) cycling $R_S(t)$, the switch resistance, between a high-resistance dark-state, $R_S(off)$, typically $> 10^7$ ohms, and a low-resistance illuminated state, $R_S(on)$, typically $< 10^2$ ohms. Errors in

accuracy (dynamic range limitations) occur in the signal data samples produced for the A/D converter. For our purposes, we will consider that the (preamplified) signal voltage ranges from $-V_o$ to $+V_o$, thereby yielding a quantization step size of $2V_o/2^n = V_o/2^{n-1}$. Analysis of these errors is considered in several categories: (1) tracking error, whereby the voltage across C_H lags V(t) because of nonzero resistance $R_o + R_S(on)$; (2) discrete electronic charge error resulting from statistical fluctuations of charge flowing onto (or from) C_H for a given V(t) time profile (with R_S closed); (3) pedestal/feedthrough error resulting from the finite turn-off time of the photoconductor switch (finite decay time of optically generated charge carriers), whereby charge will flow onto (or from) C_H after the laser (clock pulse) shuts off, including feedthrough error from charge leakage onto (or from) C_H through $R_s(off)$ after R_S reaches a maximum; (4) droop error resulting from C_H discharging through R_A (when R_S is open); (5) parasitic capacitance error resulting from the nonzero value of the switch capacitance, C_S; (6) thermal noise sources in the resistive and capacitive elements of the circuit; and (7) generation/recombination noise resulting from the generation and recombination of charge carriers in the switch.

1. Tracking Error. The sample-/track-and-hold cycle starts with the laser turning the photoconductor switch on (illuminated or closed switch) at $t = 0$ for one-half cycle of f_S (this is typical in A/D converter circuits), the sample frequency. This allows the hold capacitor, C_H, to track the signal by being charged through $R_S(on)$ and R_o. The current through C_S is negligible compared to the current through $R_S(on)$ and the turn-on response time of the switch is negligible (instantaneous) compared to the sample clock period. At the end of the track cycle, the voltage uncertainty or error, ΔV_H, across C_H must be within one-half the least significant bit (LSB) of the "true" voltage (the fractional error in charge on C_H must be less than one-half the LSB):

$$\frac{\Delta V_H}{V_H(\tau_S/2)} = e^{\frac{-\tau_S}{2RC}} \leq \frac{1}{2^{n+1}}$$

where $V_H(\tau_S/2)$ = true signal voltage across C_H at $t = \tau_S/2$ (end of tracking period), $\tau_S = 1/(f_S)$, and $RC = [R_o + R_S(on)]C_H$. One can solve for the minimum resistance, R [where $\ln 2^{n+1} = (n + 1)\ln 2 \approx 0.7(n + 1)$].

$$\boxed{[R_o + R_S(on)] \leq \frac{1}{1.4(n+1)f_S C_H}} \qquad (1)$$

2. Discrete Electronic Charge Error. For the discrete electronic charge error, one considers that the number of electrons, m, that flow through the switch in τ_S seconds obeys a Poisson distribution with an average error of $\sqrt{m}$. The corresponding voltage uncertainty, ΔV_H, must be less than one-half the LSB:

$$m_{max} = \frac{2C_H V_o}{q_o} \Rightarrow \Delta V_H\Big|_{max} = \frac{q_o\sqrt{m_{max}}}{C_H} \leq \frac{2V_o}{2^{n+1}}$$

where q_o is the electron charge.

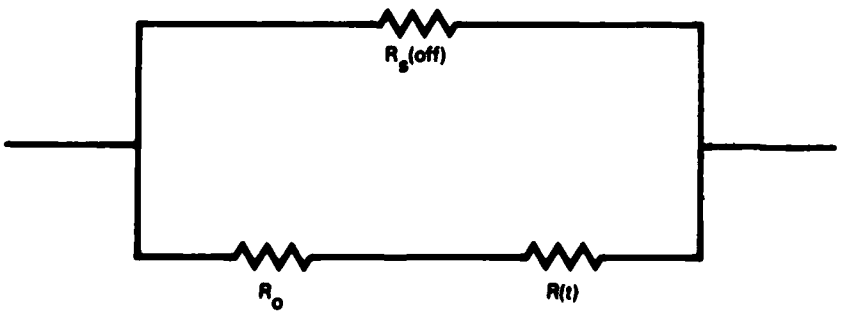
Note: During a span of τ_S seconds (from the end of one hold period to the end of the next), the greatest change in signal voltage, V(t), giving the largest flow of charge, $q_o m_{Max}$, onto C_H occurs for the highest frequency component, $f_S/2$, present in the sampled signal by an amount $2V_o$ (if the signal contains $V_o \sin \pi f_S t$). Thus

$$\boxed{C_H \geq \frac{q_o 2^{2n+1}}{V_o}} \quad . \tag{2}$$

This gives a minimum value for the hold capacitance.

3. Pedestal/Feedthrough Error. For analysis of the pedestal error, one considers that the conductivity of the switch decays exponentially when the light is turned off because of the decay of the optically generated charge carriers. This finite decay time and the noninfinite value of R_S(off) permit charge to leak from or onto C_H during the open-switch (unilluminated) period (from $t = \tau_S/2$ to $t = \tau_S$).

The resistance through which charge leaks from (or onto) C_H can be modeled according to the following schematic:



where $R_S(t) = R_S$ (on) for $0 \le t \le \tau_s/2$

$$= \left[R_S(on)e^{\frac{-\tau_S}{2\tau_o}}\right]e^{t/\tau_o} \qquad \text{for } \tau_S/2 \le t \le \tau_S \qquad \left\{\begin{array}{l}\text{repeated}\\ \text{every sample}\\ \text{cycle}\end{array}\right.$$

and τ_o = average lifetime of optically generated charge carriers where $\tau_S >> \tau_o$. The charge leakage, ΔQ_H, during the time the switch is open ($\tau_S/2 \le t \le \tau_S$) is the sum of charge leakage through each path:

$$\Delta Q_H = \int_{\tau_S/2}^{\tau_S} \frac{V_o \sin\pi f_S t dt}{R_o + R(t)} + \int_{\tau_S/2}^{\tau_S} \frac{V_o \sin\pi f_S t dt}{R_S(off)} < V_o \pi f_S \int_o^{\infty} \frac{t dt}{R_o + R_S(on)e^{t/\tau_S}}$$

$$+ \frac{V_o}{\pi f_S R_S(off)}$$

$$\Delta Q_H < \frac{V_o \pi^3 f_S \tau_o^2}{12R_o} + \frac{V_o}{\pi f_S R_S(off)}$$

where we have set $R_o = R_S$(on), and used $V_o \sin\pi f_S t \le V_o \pi f_S t$ and a simplifying time shift.

The corresponding fractional voltage change is

$$\frac{\Delta Q_H}{2V_o C_H} < \frac{\pi^3 f_S \tau_o^2}{24R_o C_H} + \frac{\tau_S}{2\pi C_H R_S(off)}$$

where each component could be set to less than one-half the LSB.

Pedestal error: $\dfrac{\pi^3 f_S \tau_o^2}{3R_o C_H} \le \dfrac{1}{2^{n-1}} \Rightarrow \boxed{\tau_o \le \dfrac{3R_o C_H}{\pi^3 f_S 2^{n-2}}}$.

Feedthrough error: $\dfrac{\tau_S}{\pi C_H R_S(off)} \le \dfrac{1}{2^n} \Rightarrow \boxed{R_S(off) \ge \dfrac{2^n \tau_S}{\pi C_H}}$.

4. Droop Error. At the beginning of the hold period (at $t = \tau_S/2$), the maximum voltage that can occur across C_H is V_o, which falls according to

$$V_H(t) = V_o e^{\frac{\tau_S/2 - t}{R_A C_H}}, \qquad (\tau_S/2 \le t \le \tau_S) .$$

The voltage across C_H at $t = \tau_S$ seconds is thus

$$V_H(\tau_S) = V_o e^{\frac{-\tau_S}{2R_A C_H}}$$

so that the maximum voltage error would be

$$\Delta V_H = V_H(\tau_S/2) - V_H(\tau_S) = V_o - V_o e^{\frac{-\tau_S}{2R_A C_H}} = V_o\left(1 - e^{\frac{-\tau_S}{2R_A C_H}}\right) .$$

The maximum fractional error in voltage across C_H after $\tau_S/2$ seconds would be

$$\frac{\Delta V_H}{2V_o} = \frac{1}{2}\left(1 - e^{\frac{-\tau_S}{2R_A C_H}} \le \frac{1}{2^{n+1}}\right) .$$

Thus

$$\boxed{R_A \ge \frac{\tau_S}{2C_H \ell n\left(\frac{2^n}{2^n - 1}\right)}} .$$

5. Parasitic Capacitance Error. When C_H has been charged by an amount Q_H (representing a signal value V_H) and the sample switch opens, some charge, Q_S, on C_H will flow onto C_S (creating a voltage V_S) because of the changing signal voltage, V(t), which will thereby change the voltage across C_H by an amount ΔV_H. Since a signal of $V(t) = V_o \sin \pi f_S t$ can change by a maximum of $\sqrt{2}V_o$ during $\tau_S/2$ seconds (the hold period), we have $V(\tau_S/2) = -V_o/\sqrt{2}$ and $V(\tau_S) = V_o/\sqrt{2}$. Thus (for perfect tracking),

$$V_H(\tau_S/2) = V(\tau_S/2) = -\frac{V_o}{\sqrt{2}} = \frac{Q_H}{C_H} \Rightarrow Q_H = \frac{-C_H V_o}{\sqrt{2}}$$

and

$$V(\tau_S) = V_H(\tau_S) + V_S(\tau_S) = \frac{Q_H - Q_S}{C_H} - \frac{Q_S}{C_S} = \frac{V_o}{\sqrt{2}}$$

$$\Rightarrow Q_S = -\sqrt{2}V_o\left(\frac{C_H C_S}{C_S + C_H}\right)$$

$$\Delta V_H = V_H(\tau_S) - V_H(\tau_S/2) = \frac{Q_H}{C_H} - \frac{Q_S}{C_H} + \frac{V_o}{\sqrt{2}} = \sqrt{2}V_o\left(\frac{C_S}{C_S + C_H}\right)$$

$$\frac{\Delta V_H}{2V_o} = \frac{1}{\sqrt{2}}\left(\frac{C_S}{C_S + C_H}\right) \leq \frac{1}{2^{n+1}}$$

$$\boxed{C_S \leq \frac{\sqrt{2}C_H}{2^{n+1} - \sqrt{2}}}.$$

6. Thermal Noise. Thermal noise in a resistive circuit element is typically represented as:

$$\langle V_{th}^2 \rangle = 4kTR\Delta f$$

where k = Boltzmann's constant; T = absolute temperature; R = resistance and Δf = noise bandwidth. This open-form expression might suggest that the noise voltage becomes infinite as bandwidth or resistance increases without bound. The truth is that this formula is only valid within some limited bandwidth beyond which the noise becomes negligibly small, because the parasitic capacitance associated with a resistive element serves as a feedback regulator (shunt capacitor) for noise voltage. For an infinitely wide bandwidth, the thermal noise voltage would be [8]

$$\langle V_{th}^2 \rangle = \frac{kT}{C}$$

where C = parasitic capacitance associated with the resistance. Typical values for our case would be T = 300°K, C = 1pf so that

$$\langle V_{th} \rangle \approx 2 \times 10^{-6} \text{ volts.}$$

Since detection of a microvolt relative to a volt represents a dynamic range of 20 bits, this noise source should present no data sample accuracy limitations.

7. Generation/Recombination (GR) Noise. This is a noise source that arises from the generation and recombination of charge carriers for which numerical values of the spectral density function of the current fluctuations typically require a value for the average number of carriers generated per second. Also needed are values for the average lifetime of a photo-excited carrier, the charge carrier drift mobility, the photoconductor switch gap length, and an effective voltage across the gap. Since these parameters can typically vary by a considerable extent, further consideration will be left to the future.

It should be mentioned in passing that the concept of shot noise (resulting from the discrete nature of electronic charge) typically depends on many of these same parameters and has been shown [8] to be related to GR noise according to

$$\frac{\langle V_{GR}^2 \rangle}{\langle V_{shot}^2 \rangle} = 2\left(\frac{\tau_s}{\tau_d}\right)$$

where τ_o = average carrier lifetime

τ_d = average carrier transit time across gap.

Typical parametric values for our application show that

$\frac{\langle V_{GR} \rangle}{\langle V_{shot} \rangle} \approx 200$, or that GR noise is the predominant contribution between these two.

The foregoing analysis is intended to be a first-cut look at constraints and limitations arising from these various circuit/switch parameters and phenomena. A more basic first-principles software model of this situation, involving solutions of simultaneous differential equations for charge and charge flow in various parts of the circuit, is being developed for a source follower configuration in which the FET gate capacitance serves as the hold capacitor. An empirical effort addressing these same issues, as well as the impact of phase noise arising from clock pulse jitter and pulse-to-pulse timing errors, is also under way.

V. CURRENT EFFORT (FY 85-86)

Present development is proceeding on several fronts:

1. The linear predictive coding technique, as described in [7] and [8], is being implemented in software for testing and analysis for suitability in this application. Critical issues include the impact of limited past signal samples and nonstationary signal statistics on prediction accuracy and stability.

2. The impact of phase noise, arising from timing pulse jitter, and mistiming on digitization accuracy is being empirically investigated by constructing a suitable audio frequency (less than 10-kHz) sample-and-hold/analog-to-digital converter system and imposing amplitude-controlled, band-limited white noise or appropriate sinusoid(s) on the clock pulses. This circuitry would utilize conventional electronic sample-and-hold switching with 16-bit A/D converters, and would create measurable random or systematic errors in sample pulse timing and pulse width (duration). These effects can be measured directly as to their impact on dynamic range limitations of the signal being digitized, and accordingly scaled to actual sample frequencies on the order of 100 MHz.

3. Development of circuitry such as that depicted in figure 11, with a state-of-the-art commercially available A/D converter (offering 6 bits of dynamic range) to operate at 100 MHz is being pursued. This will employ an InP picosecond photoconductor switch fabricated for this purpose and a suitable diode laser. Thus one can empirically address the analysis previously

described in section IV, as well as the technique employed in paragraph 2 above, at the actual sample rates ultimately to be used. This is being carried out.

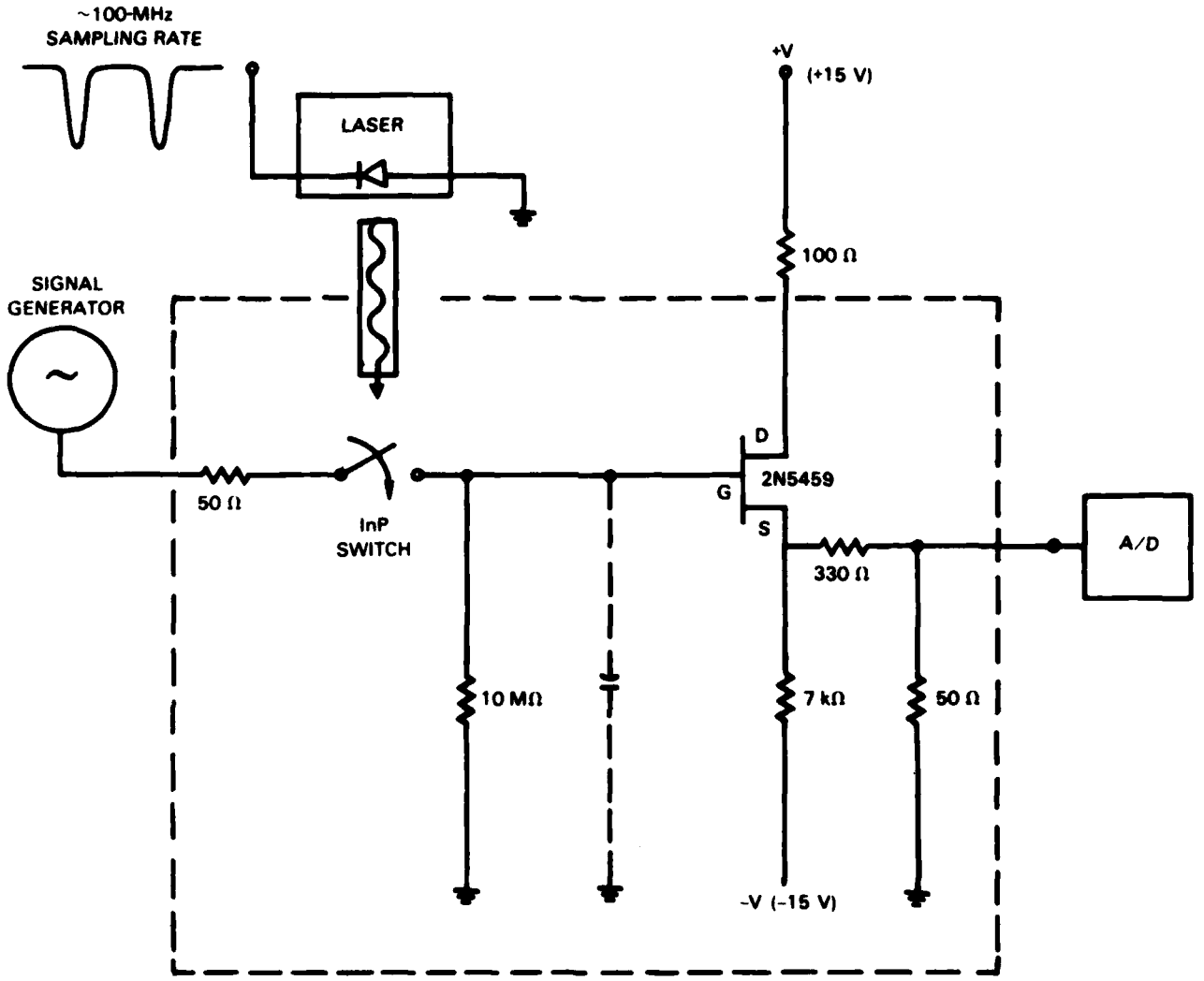


Figure 11. Track and hold circuit.

4. InP photoconductor switches, which were obtained from MIT Lincoln Laboratories in November 1983, have been found to exhibit anomalous behavior similar to that observed and reported in the fall of 1982 [8] (i.e., off-state or open-switch resistance of 100 ohms or less after initial illumination). Efforts are currently being made to understand and deal with these observed phenomena as well as the relatively low observed durability of switch lead attachment. These efforts include switch fabrication, characterization, and optimization at the NOSC microwave integrated circuit facility.

5. A version of the sample/track-and-hold circuits depicted in figures 10 and 11 is being modeled in software whereby time solutions will be obtained to the simultaneous differential equations governing charge and charge flow in this situation. Thus a more nearly exact time response can be obtained as an improvement to the analysis in the foregoing Section IV.

REFERENCES

[1] McKnight, W.H., and J.M. Speiser, "Wide Dynamic Range Analog to Digital Conversion for HFDF," NOSC report to sponsor, 28 March 1983.

[2] Markel, J.D., and A.H. Gray, Linear Prediction of Speech, Springer-Verlag, New York, 1976.

[3] Hannan, E.J., Time Series Analysis, Science Paperbacks and Methuen & Co., Ltd., 1960, pp. 20-24.

[4] Rabiner, L.R., and R.W. Schafter, Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, New Jersey, 1978, pp. 410-413.

[5] McWhirter, J.G., and T.J. Shepherd, Least Squares Lattice Algorithm for Adaptive Channel Equalization, IEE Proceedings, Vol. 130, Part F, No. 6, October 1983, pp. 532-542.

[6] Brown, J.L., Uniform Linear Prediction of Bandlimited Processes from Past Samples," IEEE Transactions on Information Theory, September 1972, pp. 662-664.

[7] Splettstosser, W., On the Prediction of Band-Limited Signals from Past Samples, Information Sciences, Vol. 28, 1982, pp. 115-130.

[8] Yariv, A., Introduction to Optical Electronics, 2nd ed. (1976), Holt, Rinehart, Winston Publishing Co.

END
DATE
FILMD
3-88
DTIC