# Nonlinear Echo Cancellation of Data Signals

OSCAR AGAZZI, DAVID G. MESSERSCHMITT, SENIOR MEMBER, IEEE, AND DAVID A. HODGES, FELLOW, IEEE

*Abstract*—**This paper describes a new technique for implementing an echo canceller for full-duplex data transmission (such as in digital subscriber loops and voiceband data sets). The canceller can operate in spite of time-invariant nonlinearities in the echo channel or in the implementation of the echo canceller itself (such as in the D/A converters). The basic structure of the linear echo canceller is not changed, but taps are simply added to account for the nonlinearity. The number of taps which must be added depends on the degree of nonlinearity which must be compensated. Numerical results based on computer simulation are given which show that typical nonlinearities encountered in MOS D/A converters can be compensated by a relatively small number of taps added to the linear echo canceller, and substantial improvement in the cancellation results.**

## I. INTRODUCTION

DATA echo cancellers have received considerable attention in recent years in connection with digital subscriber loop modems [1] and full-duplex voiceband data modems [2]. A typical configuration for a canceller in a digital subscriber loop is shown in Fig. 1 (the voiceband data modem application is similar). An inherent two-wire transmission facility is turned into an equivalent four-wire connection using a hybrid at each end. Data can then be transmitted simultaneously in both directions. However, the attenuation of the hybrid between its two four-wire inputs can be as low as approximately 10 dB. The purpose of the canceller is to remove the "near-end crosstalk" or "echo" signal which feeds through the hybrid into the local receiver, interfering with the data signal coming from a distant transmitter. Since the latter data signal may be highly attenuated (40–50 dB), the required attenuation of the echo signal is large, on the order of 50–60 dB, in order to achieve an acceptable signal-to-echo interference ratio at the receiver input for the maximum expected line attenuation.

Most canceller implementations described so far completely neglect the effect of nonlinear distortion in the echo path or in the echo replica. An exception to this uses an echo canceller with $2^N$ taps to synthesize an impulse response of $N$ samples [3]. This canceller, called a "memory compensation" or "table look-up" canceller, assigns an independent output to each possible combination of $N$ transmitted bits and, thus, is completely general as to the kind of nonlinearity that it can correct. The price paid for this generality is $2^N$ taps, rather than $N$, and a structure in which, at each sample time, only one tap weight can be updated. The consequences of this are that for large $N$ the required memory becomes very large and the adaptation very slow.

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication after presentation at the International Conference on Communications, Philadelphia, PA, June 1982. Manuscript received January 28, 1982; revised June 7, 1982.

The authors are with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720.

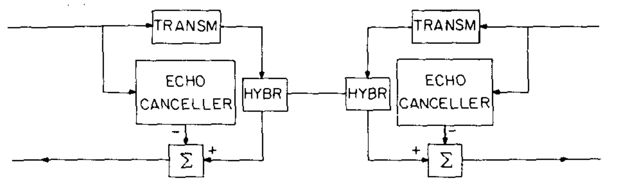



Fig. 1. Subscriber loop modems communicating on two wires. Echo cancellation techniques ensure full duplex transmission with adequate channel separation.

Achieving a 50–60 dB cancellation in a monolithic echo canceller is challenging in the face of the inherent nonlinearities in monolithic A/D and D/A converters due to processing variations and component variations. Nevertheless, these systems have to deal with small amounts of nonlinear distortion, i.e., a channel which is "almost" linear. An algorithm to allow correction of small amounts of nonlinear distortion without a large complexity penalty or adaptation speed penalty is highly desirable. In this paper we present an algorithm that is intermediate between the linear and the completely general nonlinear table look-up algorithms mentioned above. It can correct nonlinearity using extra taps, and the number of extra taps increases in direct relation to the degree of the nonlinearity that it is required to track. As a limiting case for very strong nonlinearity, $2^N$ taps are required, in which case it becomes equivalent to the table look-up algorithm [3].

As an example of the kinds of nonlinearity appearing in a practical system, consider the subscriber loop modem shown in Fig. 1. The following sources of nonlinear distortion can be identified.

1) *Transmitted Pulse Asymmetry:* When nominally balanced positive and negative pulses are transmitted, in practice there will be a slight imbalance which cannot be compensated for by a linear echo canceller. To achieve 50 dB or more of echo cancellation, a linear echo canceller would require that the uncompensated transmitted pulse asymmetry be kept below some −60 dB, which can be achieved with careful circuit design at the cost of increased complexity.

2) *Saturation in Transformers:* This will lead to a slight nonlinearity which can be controlled by choice of a bulkier transformer.

3) *Nonlinearity of Data Converters:* The echo canceller is typically implemented as a digital processor, since its input consists of an inherently digital bit stream. This suggests that the actual cancellation be done digitally, requiring A/D conversion of the received signal (containing the echo), or the cancellation can be done in analog, requiring D/A conversion of the canceller output. These data converters constitute the

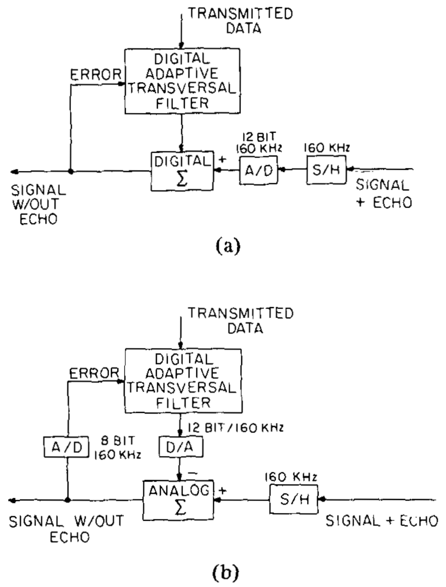



Fig. 2. Alternative echo canceller configurations. (a) Fully digital echo canceller. (b) Digital transversal filter and analog cancellation.

most important source of nonlinearity, particularly where monolithic converters without trimming are to be employed.

Examining the alternatives in the use of data converters in greater detail, Fig. 2 shows two possible configurations. In Fig. 2(a) a purely digital echo canceller using a front end A/D is considered. For a digital subscriber loop with a bit rate of 80 kbits/s and a minimum of 50 dB of echo cancellation, the A/D needs a resolution of 12 or more bits with 1/2 LSB integral linearity and a conversion time of 6 $\mu$s or less. In Fig. 2(b) the output of the echo canceller is converted to analog and the cancellation is performed in the analog domain. The error signal needed by the digital processor echo canceller is converted to digital by a lower resolution (possibly only one bit) A/D converter. The required resolution is at least 12 bits with 1/2 LSB integral linearity in the D/A and as many as 8 bits in the A/D (which needs to be monotonic, but not necessarily linear). The problem of the linearity of the data converters is a very important one in the context of an MOS monolithic implementation of these modems. Specifically, the configuration of Fig. 2(b) is particularly attractive, since D/A converters of the required speed and resolution have already been demonstrated in MOS technology [4]. However, the linearity requirements can only be achieved using self-correction or trimming, which are costly solutions. An alternative solution to the problem, in which the transversal filter summation is done by analog circuitry and, thus, the adaptation can compensate for the D/A nonlinearity, is shown in Fig. 3 and has also been demonstrated [5]. However, it cannot correct other sources of distortion like pulse asymmetry or saturation in transformers. Furthermore, digital circuits benefit more from the shrinking design rules and are easier to design than their analog counterparts, and thus a technique amenable to digital implementation like the one presented here is likely to be preferred in the future.

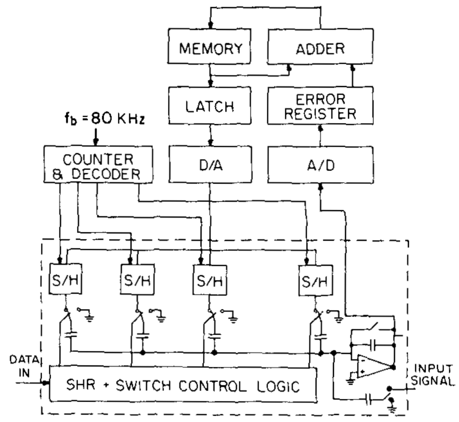



Fig. 3. Echo canceller configuration in which the D/A nonlinearity is compensated by the adaptation algorithm.

The technique described here is also interesting in other respects. It leads to a systematic design procedure for using a binary transversal filter (where the delays are implemented by shift registers) even in the context of a multilevel transmitted signal and line codes with memory. The design procedure can also take advantage of redundancies in the transmitted line code in the form of simplification of the echo canceller hardware. The technique is also applicable to the implementation of the decision feedback equalizer [6] feedback filter, which has a structure very similar to that of the echo canceller, although the requirements on this filter are relatively relaxed and compensation for nonlinearities may not be required.

In Section II a method of expanding an arbitrary nonlinear function of a number of bits in a series with a finite number of terms is presented. This expansion serves as the basis for the nonlinear echo canceller design procedures described later. Then, in Section III the application of this expansion to multilevel transmitted signals, redundancies in the line code, and nonlinearities in the echo channel and the canceller itself are considered. Section IV gives simulation results for the types of nonlinearities typically encountered in MOS D/A converters. These results indicate that, depending on the number of bits in the D/A converter, a 20 dB or greater increase in echo attenuation can be obtained by incorporating compensation for the D/A nonlinearity with a modest increase in canceller complexity. Readers desiring less detail may wish to skip Section III, concentrating instead on the results in Section IV.

## II. A BINARY SERIES EXPANSION OF A NONLINEAR FUNCTION

Let $f(B_0, \cdots, B_{N-1})$ be an arbitrary (nonlinear) function of $N$ bits, where $B_i$ assumes only two values. In the following derivation we will assume that the two values are 0 and 1. However, the expansion is also valid for other pairs of values, including the pair of values +1 and −1 which is particularly useful in many circumstances. Over all combinations of $N$ bits this function assumes a total of $2^N$ possible values (which are not necessarily distinct). We now show that this function can be represented as a series with a finite number of terms,

$$\begin{aligned} f(B_0, \cdots, B_{N-1}) &= f_0 + \sum_{k=0}^{N-1} f_1(k) B_k + \sum_{k_1 \neq k_2} f_2(k_1, k_2) B_{k_1} B_{k_2} + \cdots \\ &+ \sum_{k_1 \neq k_2 \neq \cdots \neq k_L} f_L(k_1, k_2, \cdots, k_L) B_{k_1} B_{k_2} \cdots B_{k_L} + \cdots \\ &+ \sum_{k=0}^{N-1} f_{N-1}(k) B_0 B_1 \cdots B_{k-1} B_{k+1} \cdots B_{N-1} \\ &+ f_N B_0 B_1 \cdots B_{N-1}. \end{aligned} \tag{2.1}$$

The general $L$th-order sum is over all combinations of $L$ of the $N$ indexes. Thus, for example, in the second-order term $B_1B_2$ only appears once, and not as a separate $B_2B_1$ term. To reduce the number of arguments, the subscripts of the *missing* bits have been used as arguments of the last $N/2$ coefficients $f_r(\cdot)$. The total number of terms can be obtained by observing the number of combinations of $N$ bits taken $L$ at a time is $\binom{N}{L}$. Thus, the total number of terms in the representation is

$$\sum_{L=0}^{N} \binom{N}{L} = 2^N. \tag{2.2}$$

Since there are, thus, $2^N$ free $f$ parameters in the sum of (2.1), it is not surprising that a function with $2^N$ values can be represented. When the $N$ bits in (2.1) are taken as $N$ bits out of a continuous bit stream, then the expansion of (2.1) becomes similar to a Volterra series expansion of a general nonlinear time-invariant system [7], with the important exception that only a finite number of terms is required for an exact representation.

Expansion (2.1) can be proven simply by writing a system of $2^N$ equations for the $2^N$ possible values of the nonlinear function. Since the system also has $2^N$ unknowns, it is possible, although usually difficult, to solve it. However, solving the system is particularly easy in the $-1, +1$ representation, since, in that case, the matrix of the system can be shown to be orthogonal, and its inversion becomes trivial. Furthermore, it can be shown that this matrix is a Hadamard matrix, which is important in other areas of signal processing such as image encoding [12]. These results will be given in more detail in a future paper, where other applications of expansion (2.1) will also be analyzed.

Here we give a simple proof of (2.1) by induction for the 1.0 case, and extend this proof to other pairs of values in Section II-B. Note first that

$$f_0 = f(000 \cdots 0), \tag{2.3}$$

the function evaluated for all zeros, since all the higher order terms are zero. Then, evaluating the function for a single 1 in the argument at position $k$, all the second- and higher order terms are zero and

$$f_1(k) = f(00 \cdots 010 \cdots 0) - f_0 \tag{2.4}$$

where the single 1 is in position $k$. Similarly, when the function is evaluated for two ones in positions $k_1$ and $k_2$, all the third- and higher order terms are zero and

$$\begin{aligned} f_2(k_1, k_2) &= f(0 \cdots 010 \cdots 010 \cdots 0) \\ &\quad - f_1(k_1) - f_1(k_2) - f_0. \end{aligned} \tag{2.5}$$

Proceeding by induction, all the constants in the expansion can be evaluated. Not only does this prove the result, but it also elaborates a procedure by which the constants of the expansion can actually be evaluated for a given function of $N$ bits.

Understanding of this expansion can perhaps be enhanced by a simple example. For a function of three bits $f(B_0, B_1, B_2)$ the expansion becomes

$$\begin{aligned} f(B_0, B_1, B_2) &= f_0 + f_1(0)B_0 + f_1(1)B_1 + f_1(2)B_2 \\ &+ f_2(2)B_0B_1 + f_2(1)B_0B_2 \\ &+ f_2(0)B_1B_2 + f_3B_0B_1B_2 \end{aligned} \tag{2.6}$$

where there are $2^3 = 8$ terms total. Interestingly, this expansion can be written in the form

$$\begin{aligned} f(B_0, B_1, B_2) &= f_0 + f_1(0)B_0 + B_1(f_1(1) + f_2(2)B_0) \\ &+ B_2(f_1(2) + f_2(1)B_0) \\ &+ B_1(f_2(0) + f_3B_0), \end{aligned} \tag{2.7}$$

a form which easily generalizes to the general case of $N$ bits. This latter form results in a tree of switches and adders as shown in Fig. 4(a). The leaves of the tree are the values of the constants in the expansion, and the switches closest to the leaves are closed when $B_0 = 1$ and are open when $B_0 = 0$, and similarly for the switches in the other two levels of the tree. Note that in general a number of constants in the expansion contribute to the value of the function, from a minimum of one for the all zeros case to a maximum of eight for the all ones case. A number of summations have to be evaluated to determine the function, from a minimum of zero in the all zeros case to a maximum of seven in the all ones case.

An alternate representation for the function, also requiring eight constants, is shown in Fig. 4(b). This tree also has three levels (or in general $N$ levels for $N$ bits) but in this case every branch has a switch. The convention is that the switches are shown for the $B = 0$ condition, and reverse for the $B = 1$ condition. In Fig. 4(b), when the function is evaluated one and only one path through the tree has all the switches closed. Thus, only one of the constants contributes to the function evaluation, and no summations are actually required. This method is, of course, simply a table look-up, in which the eight functional values are stored.

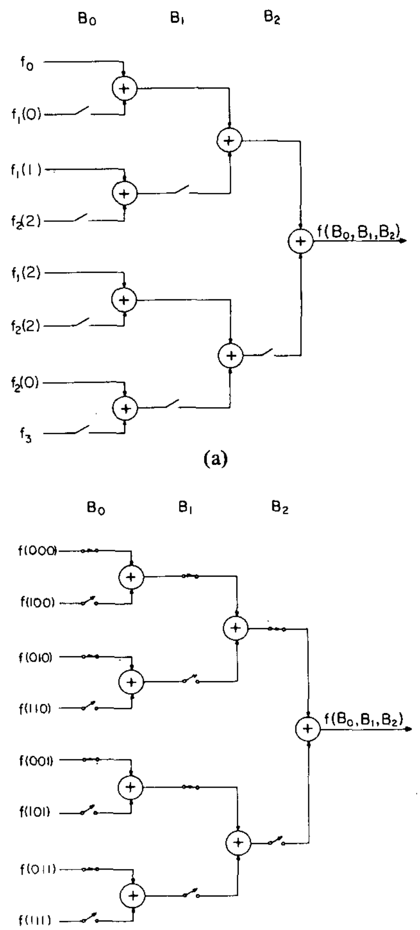



Fig. 4. Binary tree representation of nonlinear function. (a) Binary series expansion. (b) Table look-up method.

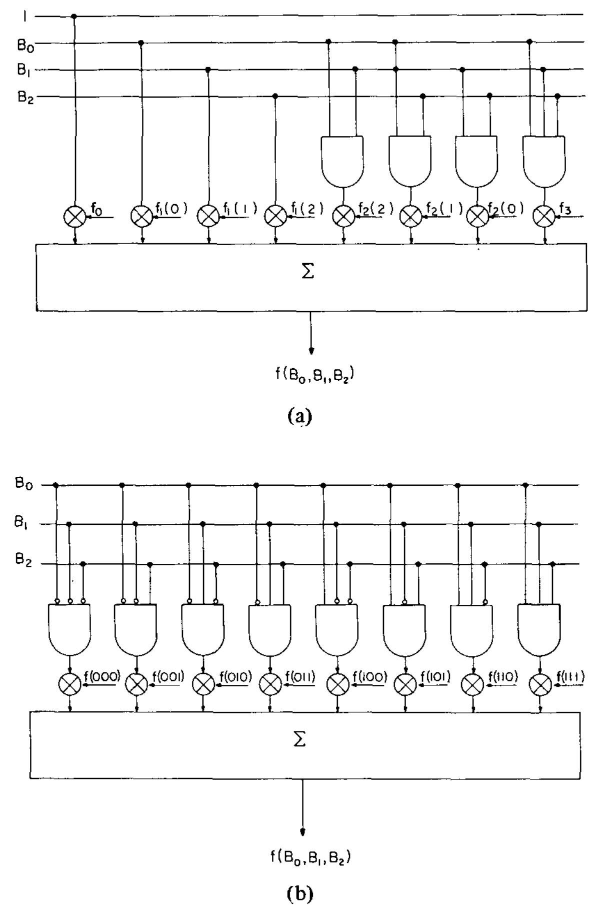



Fig. 5. Hardware implementation of nonlinear function. (a) Binary series expansion. (b) Table look-up method.

One might ask what value the expansion of Fig. 4(a) has when it requires the storage of eight constants, the same as for the method of Fig. 4(b), but unlike Fig. 4(b), it also requires summations. The answer is that in many practical situations not all of the terms in the series expansion need be retained. For example, if the function is "linear," then

$$f(B_0, B_1, B_2) = f_1(0)B_0 + f_1(1)B_1 + f_1(2)B_2 \tag{2.8}$$

and only three terms of the expansion of Fig. 4(a) need to be retained while all eight terms of the expansion of Fig. 4(b) are always required. This is of considerable importance when $N$ is large and the function is linear or nearly linear. Specifically, in most practical cases of interest, the nonlinear function $f(\cdot)$ departs only slightly from a linear function. Then, as will be shown in Section III-E, only a few of the $2^N$ terms need to be retained. It is in this property that the usefulness of (2.1) resides.

Representations of these two methods in a form more appropriate for hardware realization are shown in Fig. 5. In Fig. 5(a), note that the products of bits are easily generated using "and" gates. While the representation of Fig. 5(b) for the table look-up method does not make any sense (simply storing the values in a RAM or ROM is more reasonable), this form is conceptually valuable when the adaptive filtering application is considered in the next section.

### A. Expansion of Incompletely Specified Functions

When the function $f(B_0, \cdots, B_{N-1})$ is not specified for some particular $N$-bit sequences, a corresponding reduction in the number of terms in the expansion of (2.1) can be obtained. An application of this fact is given in Section III-A.

Suppose that the function is specified for $M < 2^N$ values of its arguments. Then, no more than $M$ terms in the expansion are required. To see this, the procedure to determine the expansion coefficients can be modified as follows. When the procedure reaches one of the $N$-bit sequences, for which the value of the function is not specified, the value of the expansion is a "don't care" for this particular argument. Therefore, the expansion coefficient being determined can be set to any arbitrary value. In particular, a value of zero effectively eliminates one of the terms of the expansion. Setting to zero the coefficients of all the terms corresponding to bit patterns for which the function is not specified results in precisely $(2^N - M)$ zero coefficients, leaving a maximum of $M$ nonzero coefficients.

A natural application is to obtain an expansion such as (2.1) for a function $f(C)$ where $C$ assumes one of $M$ values. Then, for $N$, chosen such that $2^N \geqslant M$, $M$ different $N$-bit sequences can be assigned to each of the $M$ values of $C$. An expansion of the form of (2.1) with a maximum of $M$ terms then results. This procedure will be illustrated in Section III-A for a multilevel transmission application.

### B. Expansion in Terms of Other Binary Variables

In some applications, it is desirable to obtain an expansion of the form of (2.1) in a set of variables in which each assumes two values, like $B_k$, but not the particular values 0 and 1. For example, in data transmission, it is common to transmit levels 1 and $-1$ rather than 0 and 1. The former values have the advantage, as will be seen later, of having statistical properties which are easier to handle.

Let the variable $C_k$ assume one of two values. Then, it follows from (2.1) that

$$C_k = a + bB_k \tag{2.9}$$

where $B_k$ assumes the values 0 and 1, and $a$ and $b$ are some appropriately chosen constants. It is shown in the Appendix that an expansion of the form of (2.1) can be obtained with $B_k$ replaced by $C_k$, $0 \leqslant k \leqslant N-1$. Furthermore, a procedure is given which enables one to determine the coefficients of this expansion starting from the coefficients of (2.1).

## III. APPLICATION TO ECHO CANCELLATION

The usual assumption in the design of an echo canceller for data transmission is that the echo signal consists of a linear superposition of $N$ data symbols,

$$e_k = \sum_{j=0}^{N-1} C_{k-j} h_j \tag{3.1}$$

where $e_k$ is the current echo signal, $C_k$ is the current transmitted data symbol, assuming one of $M$ possible values ($M$-ary transmission), and $h_0, \cdots, h_{N-1}$ are the impulse response samples of the echo channel. In this section we relax this linearity assumption and show how nonlinearities in the echo channel and in the echo canceller itself can be compensated in the canceller using the series expansion of (2.1). It will be shown that this method is considerably more attractive than the table look-up method [3], particularly when the number of bits $N$ is large and the nonlinearities are mild.

In Section III-A the application of this expansion to multilevel data transmission will be discussed. Then, in Section III-B the application to a nonlinear channel and/or a canceller, which for implementation reasons is nonlinear, will be explored. Section III-C explores the modifications which are desirable when typical line codes are employed. Section III-D derives an adaptive algorithm which can be used to "learn" the characteristics of the nonlinear channel and the nonlinearity of the canceller itself (this adaptation algorithm turns out to be essentially the same as for a linear canceller). Section III-E considers the truncation of expansion (2.1) to a relatively small number of terms, and describes a procedure for determining which terms to retain. Finally, Section IV will give numerical results based on computer simulations for reasonable channel and canceller models to illustrate the viability of the techniques. Readers interested in less detail are encouraged to skip to Section IV.

### A. Multilevel Transmitted Signals with Linear Canceller

It was first pointed out by Mueller [8] that the echo canceller for data transmission is particularly attractive to implement when the transmitted data bits are inputted directly to the canceller, resulting in a "binary transversal filter" in which the delay elements store individual bits rather than analog values and the need for multiples is eliminated. When the transmitted data symbols are multilevel, as is usually the case, for example, in voiceband data transmission, then this advantage would seem to be partially negated. For $M$ transmitted levels, the transversal filters require the storage of $M$ values at each stage and multiplies of the tap-weights by one of the transmitted data levels. For certain signal constellations, the latter values can be particularly inconvenient, as, for example, the square root of 2.

In the instance of multilevel data the expansion (2.1) can be used to obtain a simpler implementation. Let $L$ be an integer such that $2^L \geqslant M$. Then, the transmitted level $C_k$ can be represented as a function of $L$ bits

$$C_k = f(B_{1,k}, B_{2,k}, \cdots, B_{L,k}) \tag{3.2}$$

which in turn can be expanded as in (2.1). As shown in Section II-A, at most, $M$ terms are required in this expansion.

This result will now be illustrated for $M = 2$ through $M = 5$. For $M = 2$ level transmission, $L = 1$ and (2.1) becomes

$$C_k = a + bB_{1,k} \tag{3.3}$$

for some constants $a$ and $b$. Section II gives a procedure for finding the two constants, but in this case it is not necessary to find them since, as will be shown shortly, the adaptation mechanism of an adaptive canceller will automatically find the right constants without need for the designer to specify them.

For a three-level transmitted signal, let $L = 2$ and assign the bit patterns 00, 01, and 10 to the three levels. Then, in the expansion of (2.1) the term corresponding to the 11 bit pattern, which is $f_2$, can be set to zero, resulting in an expansion of the form

$$C_k = a + bB_{1,k} + cB_{2,k} \tag{3.4}$$

where there are three constants. Alternatively, if the bit pattern 01 is not assigned, then the $f_1(2)$ coefficient can be set to zero and

$$C_k = a + bB_{1,k} + cB_{1,k}B_{2,k} \tag{3.5}$$

which is of a slightly different form but still has three constants. Similarly, there are two other possibilities for the expansion, corresponding to not assigning the 00 or 10 bit patterns.

When the number of transmitted levels is four, the expansion of (2.1) directly becomes the form

$$C_k = a + bB_{1,k} + cB_{2,k} + dB_{1,k}B_{2,k} \tag{3.6}$$

for some constants ($a$, $b$, $c$, $d$). Finally, for $M = 5$, choose $L = 3$, and assign the bit patterns 000, 001, 010, 100, and 011 to the five levels. Then, the expansion is of the form

$$C_k = a + bB_{1,k} + cB_{2,k} + dB_{3,k} + eB_{2,k}B_{3,k} \tag{3.7}$$

where there are, in this case, 55 other ways in which the five levels can be assigned to patterns of three bits, each resulting in a different form of the expansion.

It should be emphasized that in any of these illustrative expansions one or more of the constants can be zero. In fact, one criterion for choosing from among the possible expansions is the number of nonzero terms which result for the particular transmitted levels.

Using these expansions, the received echo signal of (3.1) can be represented in a different form, in which the terms are represented in terms of binary rather than $M$-level data signals. For example, for the four-level signal of (3.6), (3.1) becomes

$$e_k = \sum_{j=0}^{N-1} h_j(a + bB_{1,k-j} + cB_{2,k-j} + dB_{1,k-j}B_{2,k-j}), \tag{3.8}$$

a representation of which is shown in Fig. 6. The echo response is represented as a transversal filter with $3N + 1$ taps, each of which needs to store only a single bit. The delay line can thus be implemented by a shift register as in the binary transmission case, and the tap-weights do not require multiplies. The equivalent echo impulse response $(g_0, \cdots, g_{3N+1})$ is a function of the actual channel impulse response as well as the constants $(a, b, c, d)$. If an adaptive echo canceller is constructed from the model of Fig. 6, there is no need to explicitly incorporate these latter constants into the design, since the adaptation mechanism will automatically incorporate them. This should become clearer in Section III-E where adaptive filtering in the context of the expansion (2.1) will be elaborated.

When the transmission is two-level, then only one of the three shift registers is required, so that there are $N + 1$ total taps. For three-level transmission, only two of the shift registers are required, so that there are $2N + 1$ total taps.

In general, for $M$-level transmission, the structure of a multiply-free binary transversal filter can be retained and the details of the multiple level transmitted signal can be left to the adaptation mechanism to sort out. In each case, a maximum of $(M - 1)N + 1$ taps are required in the binary transversal filter. This technique has two advantages. First, the implementation is simplified by incorporating the details of the multilevel signal into the tap weights. Second, in practice there will be some uncertainty in the transmitted levels due to component tolerances, etc., for which the canceller will automatically compensate. For example, a mismatch between a positive and negative transmitted level will have no adverse effect on the echo attenuation which can be achieved.

### B. Nonlinear Channel with Nonlinear Canceller

The most interesting application of the expansion of Section II is to the compensation of nonlinear as well as linear effects in the channel, as well as in the canceller itself. The method by which this can be done will be considered in this section.

Assume that the echo signal is not a linear superposition of data digits as in (3.1), but rather that the echo is a general non-

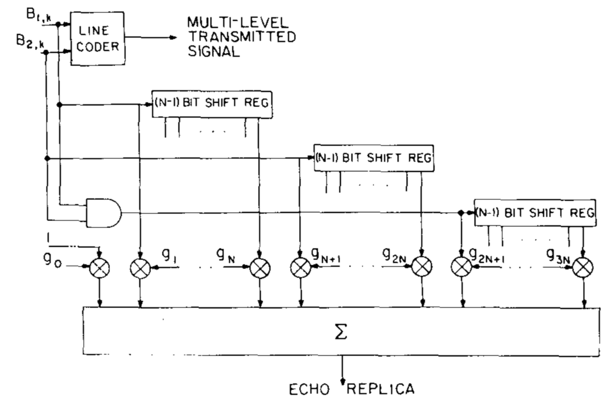



Fig. 6. Linear echo canceller for multilevel transmitted signals.

linear function of the current and past $N - 1$ data digits,

$$e_k = f(C_k, C_{k-1}, \cdots, C_{k-N+1}). \tag{3.9}$$

This model precludes the possibility of the function $f$ being a function of $k$ and, thus, requires that the nonlinear echo channel be time invariant. This is the same assumption required for the Volterra series representation of a nonlinear system [7]. However, when the canceller is made adaptive as in Section III-E, the canceller can compensate for a nonlinear echo channel which is a slowly varying function of time.

Further assume, for simplicity, that the data digits are binary, assuming one of two values. As was mentioned in Section II-B, the expansion of (2.1) is valid for an arbitrary two-level signal $C_k$ as well as for a signal $B_k$ which assumes the values 0 and 1. It is convenient to write this expansion in a vector inner product notation. Toward this end, define a $2^N$-dimensional "augmented transmitted data vector"

$$\begin{aligned} \mathbf{c}_k = (1, C_k, \cdots, C_{k-N+1}, C_kC_{k-1}, C_kC_{k-2}, \cdots, \\ C_{k-N+2}C_{k-N+1}, \cdots, C_kC_{k-1} \cdots C_{k-N+1})^T \end{aligned} \tag{3.10}$$

where each term in the series representation of (2.1) is represented and the superscript $T$ denotes transpose. The subscript $k$ on $\mathbf{c}_k$ reflects the fact that this vector is changing with time in accordance with the current and last $N - 1$ bits of the data sequence.

In a similar way, define a $2^N$-dimensional "augmented echo path vector"

$$\begin{aligned} \mathbf{g} = (g_0, g_1(0), g_1(1), \cdots, g_1(N-1), g_2(0, 1), \cdots, \\ g_2(N-2, N-1), \cdots, g_N)^T \end{aligned} \tag{3.11}$$

which is a vector of coefficients of an expansion of the form (2.1) and, in accordance with (3.9), is not a function of $k$. Then, a more compact notation for expansion (2.1) is as an inner product of an augmented data vector with the augmented echo path vector

$$e_k = \mathbf{c}_k^T \cdot \mathbf{g}. \tag{3.12}$$

It is natural to assume that the canceller implements an ex-

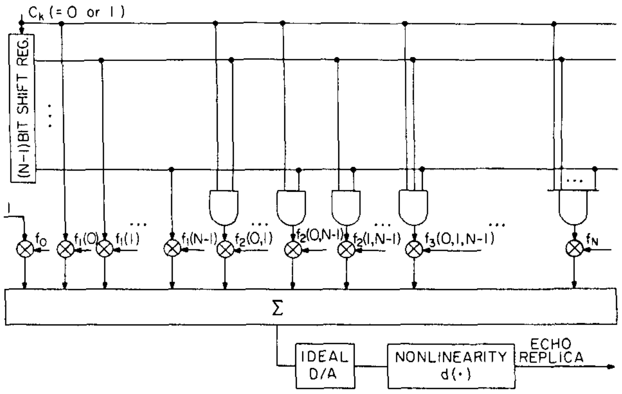



Fig. 7. Nonlinear echo canceller for nonlinear channel. Distortion introduced by the channel and by the D/A nonlinearity $d(\cdot)$ are both compensated.

pansion of the form of (3.12) with tap vector $\boldsymbol{a}$,

$$\boldsymbol{a} = (a_0, a_1(0), a_1(1), \cdots, a_1(N-1), a_2(0, 1), \cdots,$$
$$a_2(N-2, N-1), \cdots, a_N) \tag{3.13}$$

so that an arbitrary nonlinear echo can be exactly cancelled. A hardware realization of this canceller is shown in Fig. 7 for the case where $C_k$ assumes the values 0 and 1. Also included in Fig. 7 is a nonlinearity $d(\cdot)$ which models the undesired but unavoidable nonlinearity of the D/A converter at the canceller output in Figure 2(a). This nonlinearity follows an ideal D/A converter. Ignoring the quantization due to the D/A converter, the echo replica can be written as

$$\hat{e}_k = d(\boldsymbol{c}_k{}^T \cdot \boldsymbol{a}). \tag{3.14}$$

The interesting question which arises is whether the incorporation of the augmented transmitted data vector into the canceller can compensate for the (D/A) nonlinearity $d(\cdot)$ as well as the echo channel nonlinearity. To answer this question, note that since (3.14) is a (nonlinear) function of $N$ bits, it follows from Section II-B that there exists a $2^N$-dimensional vector-valued function $\boldsymbol{D}[\boldsymbol{a}]$ of a $2^N$-dimensional vector $\boldsymbol{a}$ such that

$$d[\boldsymbol{c}_k{}^T \cdot \boldsymbol{a}] = \boldsymbol{c}_k{}^T \cdot \boldsymbol{D}[\boldsymbol{a}] \tag{3.15}$$

where $\boldsymbol{D}[\boldsymbol{a}]$ is a $2^N$-dimensional vector-valued nonlinear transformation induced by the nonlinear function $d(\cdot)$ on the coefficient vector $\boldsymbol{a}$. Note that this relation is still linear in the augmented transmitted signal vector. As long as a vector $\boldsymbol{a}$ can be found such that

$$\boldsymbol{c}_k{}^T \cdot \boldsymbol{g} = \boldsymbol{c}_k{}^T \cdot \boldsymbol{D}[\boldsymbol{a}] = d(\boldsymbol{c}_k{}^T \cdot \boldsymbol{a}) \tag{3.16}$$

for every signal vector $\boldsymbol{c}_k$, then $e_k = \hat{e}_k$ and, in principle, the echo canceller can cancel the echo completely even in the face of the nonlinearities. A simple sufficient condition for (3.16) to be possible is if the inverse D/A nonlinearity $d^{-1}(\cdot)$ exists, since then (3.16) becomes

$$\boldsymbol{c}_k{}^T \cdot \boldsymbol{a} = d^{-1}(\boldsymbol{c}_k{}^T \cdot \boldsymbol{g}). \tag{3.17}$$

Since the right side of (3.17) is a function of $N$ bits, Section II-B guarantees the existence of a vector $\boldsymbol{a}$ satisfying (3.17) and, furthermore, gives a procedure for finding it. It is interesting to note from (3.17) that even when the echo channel is linear (all but the $N$ linear taps of $\boldsymbol{g}$ are zero), the canceller needs more than the $N$ linear taps in order to compensate for the D/A nonlinearity.

The addition of extra nonlinear taps should partially or entirely mitigate the effects of D/A nonlinearity, allowing the full resolution of the D/A to be useful. There are monolithic D/A converter realizations which are inherently monotonic, which is sufficient for the existence of $d^{-1}(\cdot)$. Of course, in practice the quantization due to the D/A converter will prevent an exact cancellation of the echo.

The conclusion is that a linear canceller algorithm can still be used in the face of a nonlinear channel and nonlinear canceller implementation. What is necessary is to augment the $N$ bits, which are input to the canceller by the remaining bits in the augmented signal vector, resulting in a nonlinear canceller with $2^N$ taps. Of course, the hope is that considerably fewer taps than this will be required in practice.

### *C. Line Codes with Memory*

It is often desirable to use line codes which incorporate memory for the purpose of limiting dc wander, RFI, crosstalk, etc. A common example is the "bipolar" or "alternate mark inversion" line code, in which a binary signal is transmitted as a three-level signal. Each input data bit $B_k = 0$ is transmitted as $C_k = 0$, while $B_k = 1$ is transmitted alternately as $C_k = -1$ and $+1$. We will use this example to illustrate how the presence of a line code can be incorporated into the canceller design.

Section III-A showed how the three-level $C_k$ could be represented by two bits $B_{1,k}$ and $B_{2,k}$, and the transmitted level could be represented as in (3.4). In the presence of a nonlinear echo channel, the received echo signal of (3.9) can be rewritten as a function of $2N$ bits, and an expansion with $2^{2N}$ terms results. However, due to the fact that the signal is three-level, and due to the redundancy in the line code, many of these terms are unnecessary. For example, since $B_{1,k-l} = B_{2,k-l} = 1$ can never occur in the transmitted signal, all terms in the expansion containing the product $B_{1,k-l}B_{2,k-l}$, $0 \leqslant l \leqslant N-1$ can be eliminated. This will reduce the number of terms in the expansion to $3^N$. In addition, the memory in the line code will reduce the number of terms further. For example, since $C_k = 1$ cannot be preceded by $C_{k-1} = 1$, and similarly for $-1$, the terms $B_{1,k-1}B_{1,k}$ and $B_{2,k-1}B_{2,k}$ can be eliminated. Elimination of all terms of this type will, of course, reduce the total number of terms to $2^N$, the number of possible input data sequences.

The fact that we end up with $2^N$ terms by such a cumbersome procedure suggests that there must be an easier approach, and indeed there is. The bipolar encoding can be accomplished by the circuit shown in Fig. 8 [9]. A modulo 2 accumulation of all the past input bits is first performed, resulting in the binary variable $A_k$. The three-level $C_k$ is obtained by taking the difference of successive $A_k$. All we have to do is input to the canceller $A_k$ rather than $C_k$, since the linear first difference

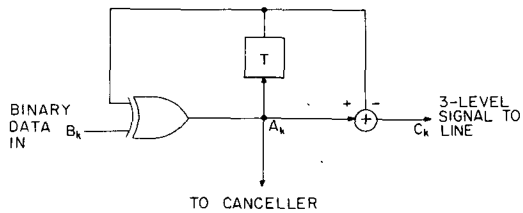



Fig. 8. Bipolar encoder.

filter can then be thought of as being a part of the echo channel and can be easily compensated by the linear taps of the canceller. When the input data sequence is independent and equally likely, the $A_k$ are likewise independent and equally likely, and the operation of the canceller, even when adaptive, is not adversely affected. The canceller will require $2^N$ taps as before, but with a lot less effort!

*D. Adaptation Algorithm*

In this section we show how the usual LMS adaptation algorithm can also be used in the presence of nonlinear distortion. From Fig. 9, the residual signal after echo cancellation is

$$r_k = e_k + s_k - \hat{e}_k + n_k \tag{3.18}$$

where $s_k$ is the data signal coming from the remote transmitter and $n_k$ is the noise term, and both are assumed to be uncorrelated with the echo. If the data digits $C_k$ are assumed to be uncorrelated and assume the values $+1$ and $-1$ with equal probability, it is easily verified that the elements of vector $\boldsymbol{c}_k$ are uncorrelated (although not independent). Then, by an analysis similar to that in [10], the mean-squared residual can be calculated to be

$$\rho = E(r_k^2) = (\boldsymbol{g} - \boldsymbol{D}(\boldsymbol{a}))^T \cdot (\boldsymbol{g} - \boldsymbol{D}(\boldsymbol{a})) + U \tag{3.19}$$

where

$$U = E(s_k^2) + E(n_k^2) \tag{3.20}$$

is the total power of the remote data signal and noise.

Assume initially that the canceller does not have a nonlinear D/A, so that $d(\cdot)$ is the identity function. Then, $\rho$ of (3.19) is quadratic, and there is a unique global minimum which can be determined by setting the gradient of $\rho$ with respect to the tap vector $\boldsymbol{a}$ to zero. As in [10] this becomes

$$\text{grad}\,\rho = -2(\boldsymbol{g} - \boldsymbol{a}) = -2E(r_k \boldsymbol{c}_k) \tag{3.21}$$

and the minimum $\rho$ occurs as expected for equal augmented echo path vector and canceller tap vector, $\boldsymbol{a} = \boldsymbol{g}$. To find this minimum adaptively, let the canceller tap vector $\boldsymbol{a}$ be a function of time $\boldsymbol{a}_k$ and use the standard gradient algorithm

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + 2\alpha r_k \boldsymbol{c}_k. \tag{3.22}$$

This algorithm is illustrated in Fig. 9 for just one tap of the canceller. As usual, the parameter $\alpha$ is adjusted to obtain the desired tradeoff between convergence rate and asymptotic

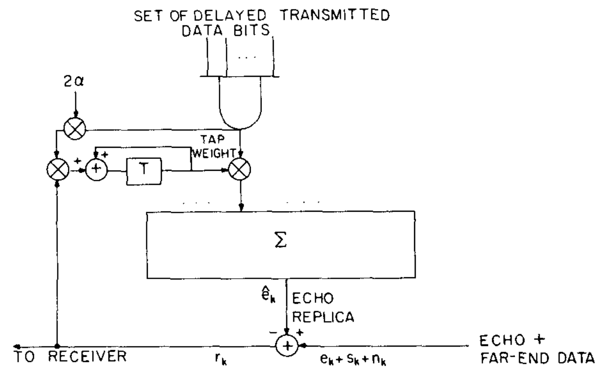



Fig. 9. Adaptation of tap weight coefficients is the same as for the linear canceller.

excess mean-square error. This is the same adaptation algorithm used in [10], and it is interesting that the presence of the nonlinearities in the channel has not affected the adaptation algorithm at all, aside from the augmentation of the transmitted signal vector with nonlinear taps.

If there is a nonlinearity $d(\cdot)$ in the echo replica path, the gradient of (3.21) becomes

$$\text{grad}\,\rho = -2\left[\frac{\partial \boldsymbol{D}(\boldsymbol{a})}{\partial \boldsymbol{a}}\right] \cdot (\boldsymbol{g} - \boldsymbol{D}[\boldsymbol{a}]). \tag{3.23}$$

If the Jacobian matrix $\partial \boldsymbol{D}(\boldsymbol{a})/\partial \boldsymbol{a}$ is nonsingular, the unique minimum of $\rho$ is obtained for

$$\boldsymbol{D}[\boldsymbol{a}] = \boldsymbol{g} \tag{3.24}$$

as in (3.16). In this case, the gradient technique will also apply, since there will be no local minima in which the algorithm can get lost. The condition of nonsingularity of the matrix $\partial \boldsymbol{D}(\boldsymbol{a})/\partial \boldsymbol{a}$ is a very mild one, since for the case of small nonlinearity (the case of interest here), $\partial \boldsymbol{D}(\boldsymbol{a})/\partial \boldsymbol{a}$ differs from the identity matrix by only a small perturbation, and the small size of the perturbation ensures that the matrix is nonsingular. Further, since the function $\boldsymbol{D}(\boldsymbol{a})$ is not known by the adaptation algorithm, it is necessary that $\partial \boldsymbol{D}(\boldsymbol{a})/\partial \boldsymbol{a}$ be replaced by the identity matrix. This is again justified for small perturbations from linearity, and results again in the standard LMS gradient algorithm of (3.22).

The speed of convergence and asymptotic residual echo can be predicted from the analysis of [10]. Although the elements of the vector $\boldsymbol{c}_k$ are not statistically independent, they are uncorrelated and zero-mean for the case where the data digits $C_k$ assume the values $+1$ and $-1$ with equal probability and are statistically independent. This condition is sufficient for the validity of the convergence analysis of [10], yielding a ratio of asymptotic residual echo to uncancellable signal of

$$\frac{E[e_k - \hat{e}_k]^2}{U} = \frac{\alpha L}{1 - \alpha L} \approx \alpha L. \tag{3.25}$$

In (3.25) the number of elements in the augmented transmitted signal vector has been assumed to be $L$ in anticipation of using only a small number $L$ of the $2^N$ taps. Hence, in general $L \ll 2^N$ will be chosen, and the asymptotic error will be correspondingly smaller than for the table look-up canceller, where $L = 2^N$. Similarly, the speed of convergence can be measured by the time constant of convergence, which is [10]

$$\tau = -\frac{1}{\log_e (1 - 4\alpha + 4\alpha^2 L)} \approx \frac{1}{4\alpha}. \tag{3.26}$$

The approximations in (3.25) and (3.26) are valid for practical values of $\alpha$, which are very small. We can compare the convergence of the linear canceller, the nonlinear canceller proposed here, and the table look-up canceller by setting the asymptotic residual errors of (3.25) equal for the three cases and then comparing the time constants of (3.26). The result is that the time constant of the nonlinear canceller proposed here is $L/N$ times as great as for the linear canceller, while it is $2^N/N$ times as great for the table look-up canceller. Thus, we pay a convergence time penalty for the extra nonlinear taps (about a factor of two for the numerical examples of Section IV), but a much larger penalty for the table look-up canceller.

### *E. Truncation of the Series Expansion*

In the preceding analysis, the full $2^N$-tap echo canceller has been considered. We expect that under the conditions of a) small nonlinearity and b) rapidly decaying echo path impulse response, most of the coefficients of vector $\boldsymbol{a}$ are negligible and can be ignored. This will be established in Section IV by simulation for typical nonlinearities encountered in MOS D/A converters. However, it is important to develop a methodology by which the nonnegligible taps can be predicted, in order to develop insight and to avoid an inordinate number of simulations.

Which coefficients need to be retained depends on the shape of the echo path impulse response and the nonlinearity. Knowing them, it is fairly straightforward to predict what taps are necessary in the transversal filter, as we will show by example in this section.

The examples we give here are for memoryless nonlinearities. A simple example of a nonmemoryless nonlinearity, where a small number of nonlinear taps is required, is transmitted pulse asymmetry in bipolar transmission [13].

If only $L$ taps are used, it is apparent that the $L$ taps which are largest in absolute value should be chosen. This is confirmed in (3.19), for when $\boldsymbol{D(a)}$ is constrained to have only $L$ nonzero elements, $\rho$ will be minimized by choosing those elements for which $\boldsymbol{g}$ is largest in absolute value. For small deviations from linearity, this will be the same as choosing the same $L$ elements of $\boldsymbol{a}$ to be nonzero.

If the characteristics of the echo channel nonlinearity and D/A nonlinearity are known and are fairly reproducible, then the taps which are important can be predicted. This will be illustrated by example. Suppose the echo channel can be modeled by an FIR filter followed by a memoryless nonlinearity $q(\cdot)$. Then, (3.1) becomes instead

$$e_k = q\left(\sum_{j=0}^{N-1} h_j C_{k-j}\right). \tag{3.27}$$

Then, the function $q(\cdot)$ can be expanded in or at least approximated by a Taylor series expansion. Consider, for example, the square term in this expansion, which becomes

$$\left(\sum_{j=0}^{N-1} h_j C_{k-j}\right)^2 = \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} h_{j_1} h_{j_2} C_{k-j_1} C_{k-j_2} \tag{3.28}$$

which can be simplified by eliminating the duplicated terms and noting from Section II-B that since ${C_{k-j}}^2$ is a binary function it can be represented as

$${C_k}^2 = a + bC_k \tag{3.29}$$

for some constants $a$ and $b$. Then, (3.28) becomes

$$\left(\sum_{j=0}^{N-1} h_j C_{k-j}\right)^2 = \sum_{j_1=0}^{N-2} \sum_{j_2=j_1+1}^{N-1} 2h_{j_1} h_{j_2} C_{k-j_1} C_{k-j_2} + a \sum_{j=0}^{N-1} h_j^2 + b \sum_{j=0}^{N-1} h_j^2 C_{k-j}. \tag{3.30}$$

From this relationship note that this square term contributes primarily to the second-order terms in (2.1), but also to the first-order term. Also, note that the important terms will generally be those for which $h_{j_1}$ and $h_{j_2}$ are both large. From this one can conclude more generally that large $n$th-order terms in $q(\cdot)$ will contribute most heavily to $n$th-order terms in the expansion of (2.1), and that generally, the large nonlinear taps will be those containing $C_{k-j}$ corresponding to the larger $h_j$.

When the D/A is nonlinear, and $d^{-1}(\cdot)$ must be incorporated, it can be expanded in a Taylor series and a similar analysis can be applied to (3.17) to determine which taps in $\boldsymbol{a}$ are important. There are at least two methods for obtaining the Taylor series expansion for $d^{-1}(\cdot)$. One method is to first find an analytical expression for $d^{-1}(\cdot)$, and then expand it in the Taylor series. The second method is to do a Taylor series expansion of $d(\cdot)$, and then use the method described in [11, p. 362] to directly find the Taylor series of its inverse. This latter procedure will be illustrated in Section IV.

Note that the validity of the expansion of (2.1) to echo cancellation does not depend on the existence of a Taylor series expansion of $d(\cdot)$ or $d^{-1}(\cdot)$, as can be seen from (3.17). When the function $d(\cdot)$ is not analytic and a Taylor series does not exist (as, for example, when the function is piecewise linear), then it can be approximated to any desired accuracy by a polynomial (which is a truncated Taylor series) and we can proceed as before.

## IV. CANCELLATION MSE WITH MOS D/A CONVERTER

In this section we study, using computer simulations, the performance of a nonlinear echo canceller in the configuration of Fig. 2(b). The operation of the adaptive echo canceller derived in Section III-E was simulated in the presence of certain nonlinearities inherent in MOS D/A converters. The purpose of simulating the canceller, rather than using the procedure described in Section II for finding the coefficients of the ex-

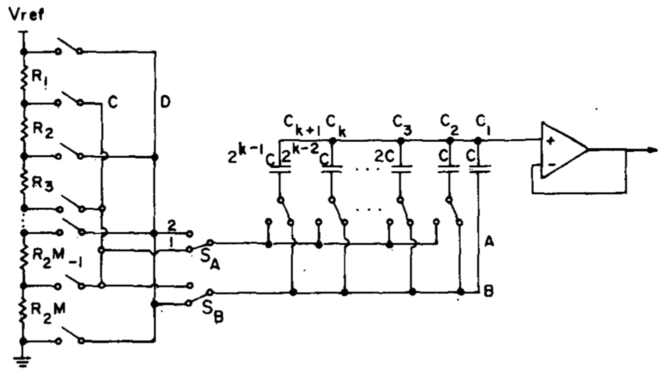
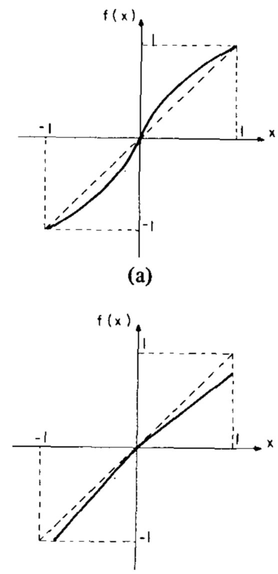



Fig. 10. DAC using resistor string and capacitor array. (Switch control is omitted for simplicity.)



(a)

(b)

Fig. 11. Typical DAC transfer functions. (a) $f(x) = 1.01333x - 0.01333x^3$. (b) $f(x) = x - 0.005/x/$.

pansion, was to establish that the adaptive algorithm does indeed work properly in the presence of nonlinearities. The asymptotic mean-square echo cancellation residual error was noted as a function of the total number of taps implemented in the canceller. The particular taps which were implemented, and the order in which they were added, was determined by first running a program which calculates all the coefficients of the expansion in accordance with the procedure of Section II for the particular nonlinearity being studied. The taps were then added in the order of decreasing absolute value.

In order to make the numerical examples realistic, assume the D/A converter is to be implemented in MOS technology using the technique shown in Fig. 10. The four most significant bits are provided by a string of 16 diffused resistors and the remaining bits (from 6 to 9 in our simulations) by a binary weighted capacitor array. Because of diffusion concentration gradients, voltage coefficient, and photolithographic mismatches, the resistors cannot be guaranteed to be equal to within 1 LSB unless laser trimming is used. Thus, in the absence of trimming, a nonlinear transfer characteristic results. This nonlinearity can have a systematic component due to concentration gradients, and a random component due to photolithographic mismatches.

Two of the most common kinds of systematic nonlinearity are shown in Fig. 11. We model the transfer characteristic of Fig. 11(a) by

$$d(x) = ax + bx^3 \tag{4.1}$$

where $a = 1.01333$ and $b = -0.01333$, and the one in Fig. 11(b) by

$$d(x) = x + b\,|x| \tag{4.2}$$

where $b = -0.005$. For the characteristic of (4.1), the nature of the inverse can be determined by finding a power series expansion for $d^{-1}(\cdot)$. Defining this power series as

$$d^{-1}(y) = \sum_{n=0}^{\infty} b_n y^n \tag{4.3}$$

then, we obtain

$$d^{-1}(d(x)) = x = \sum_{n=0}^{\infty} b_n(ax + bx^3)^n. \tag{4.4}$$

Equating coefficients in (4.4) and solving for the $b_n$'s,

$$d^{-1}(x) = \frac{1}{a}x - \frac{b}{a^4}x^3 + \frac{3b^2}{a^7}x^5 - \cdots \tag{4.5}$$

and we see that the third-order nonlinearity predominates in $d^{-1}(\cdot)$ as it does in $d(\cdot)$. The even harmonics are missing since the characteristic of (4.1) has odd symmetry about the origin. Thus, we expect that the important terms in the Volterra series expansion will be the first- and third-order terms. For the characteristic of (4.2), the easiest method is to find $d^{-1}(\cdot)$ directly and approximate it by a polynomial. Since the nonlinear portion of this nonlinearity has even symmetry, the odd powers will be missing, and the important terms in the Volterra series will be the first- and second-order terms. These conclusions are confirmed by the simulations which follow.

The simple echo path impulse response assumed in all cases was

$$g_k = e^{-0.8k}, \qquad k \geqslant 0. \tag{4.6}$$

The effect of quantization was also included. Runs with 10, 11, 12, and 13 bits and with infinite resolution were done, varying the number of taps from 1 to 26 in the case of Fig. 11(a) and from 1 to 17 for Fig. 11(b). As previously mentioned, the taps were added in the order of decreasing absolute value as determined by another program. The resulting order is shown in Table I(a) and (b), together with the residual cancellation error in Fig. 12(a) and (b). Observe that in both cases, many higher order taps are more important than the linear taps beyond the tenth. The importance of the nonlinear taps depends on the number of bits of quantization. With 10 bits there is no point to using nonlinear taps in Fig. 11(a), whereas in Fig. 11(b) the nonlinear taps give about a 10 dB reduction in asymptotic residual error. For 13 bits of resolution, with a modest number of nonlinear taps a 20-30 dB improvement can be obtained. In both cases, the number of taps is dramat-

TABLE I
(a) ORDER IN WHICH TAPS WERE INCLUDED IN THE SIMULATION OF FIG. 12(a) AND THEIR NUMERICAL VALUE. (b) ORDER IN WHICH TAPS WERE INCLUDED IN THE SIMULATION OF FIG. 12(b) AND THEIR NUMERICAL VALUE.

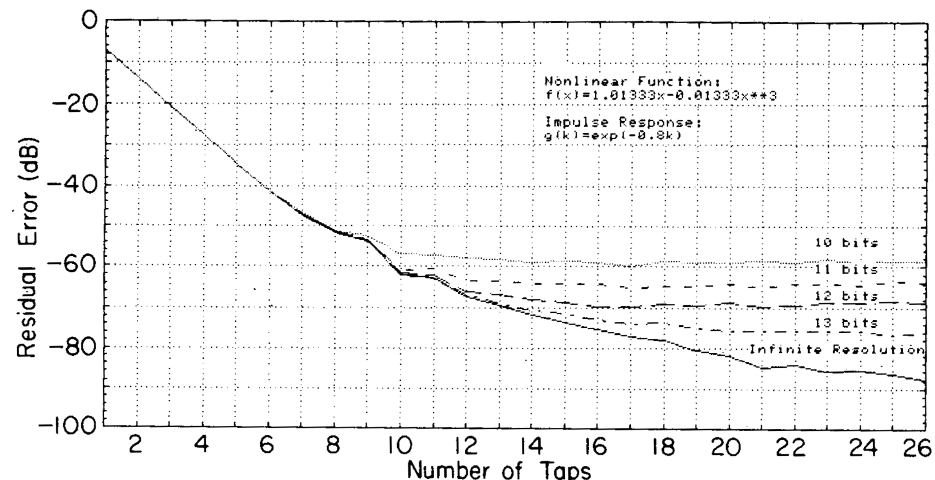
| $f(x)=1.01333x-0.01333x^3$ | | | | | |
|---|---|---|---|---|---|
| TAP NUMBER | PRODUCT TERM | NUMERICAL VALUE(*) | TAP NUMBER | PRODUCT TERM | NUMERICAL VALUE(*) |
| 0 | C0 | 0.4454855025 | 13 | C1C2C3 | 0.0000619562 |
| 1 | C1 | 0.2010037899 | 14 | C0C2C4 | 0.0000592478 |
| 2 | C2 | 0.0904013440 | 15 | C0C1C5 | 0.0000589880 |
| 3 | C3 | 0.0406264402 | 16 | C1C2C4 | 0.0000282178 |
| 4 | C4 | 0.0182554312 | 17 | C0C3C4 | 0.0000270387 |
| 5 | C5 | 0.0082029458 | 18 | C0C2C5 | 0.0000269134 |
| 6 | C6 | 0.0036856988 | 19 | C0C1C6 | 0.0000267841 |
| 7 | C7 | 0.0016559258 | 20 | C1C2C5 | 0.0000125894 |
| 8 | C0C1C2 | 0.0009801916 | 21 | C1C3C4 | 0.0000124928 |
| 9 | C8 | 0.0007441015 | 22 | C0C2C6 | 0.0000121263 |
| 10 | C0C1C3 | 0.0002906728 | 23 | C0C1C7 | 0.0000117915 |
| 11 | C0C2C3 | 0.0001319754 | 24 | C0C3C5 | 0.0000116926 |
| 12 | C0C1C4 | 0.0001308451 | 25 | C9 | -0.0000000888 |

(a)

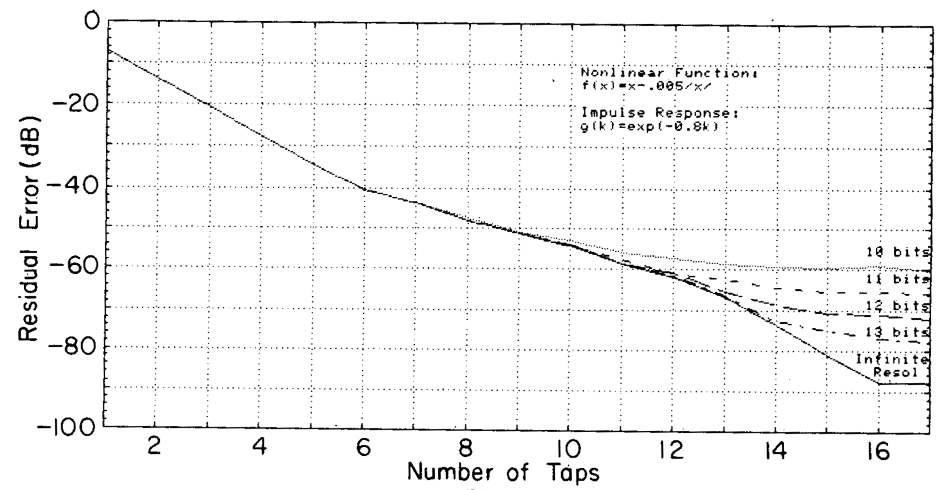
| $f(x)=x-0.005/x/$ | | | | | |
|---|---|---|---|---|---|
| TAP NUMBER | PRODUCT TERM | NUMERICAL VALUE(*) | TAP NUMBER | PRODUCT TERM | NUMERICAL VALUE(*) |
| 0 | C0 | 0.4493286908 | 9 | C0C1 | 0.0010097334 |
| 1 | C1 | 0.2018965483 | 10 | C8 | 0.0007465757 |
| 2 | C2 | 0.0907173827 | 11 | C0C2 | 0.0004535968 |
| 3 | C3 | 0.0407625660 | 12 | C9 | 0.0003355653 |
| 4 | C4 | 0.0183155797 | 13 | C0C3 | 0.0002040363 |
| 5 | C5 | 0.0082295639 | 14 | C0C4 | 0.0000915387 |
| 6 | C6 | 0.0036979516 | 15 | C0C5 | 0.0000415029 |
| 7 | 1 | 0.0022468669 | 16 | C1C3 | 0.0000001015 |
| 8 | C7 | 0.0016615218 | | | |

(*)Numerical values shown are for the infinite resolution case.

(b)




(a)




(b)

Fig. 12. Residual error as a function of the number of taps. The order in which taps are introduced is shown in Table I(a) and (b).

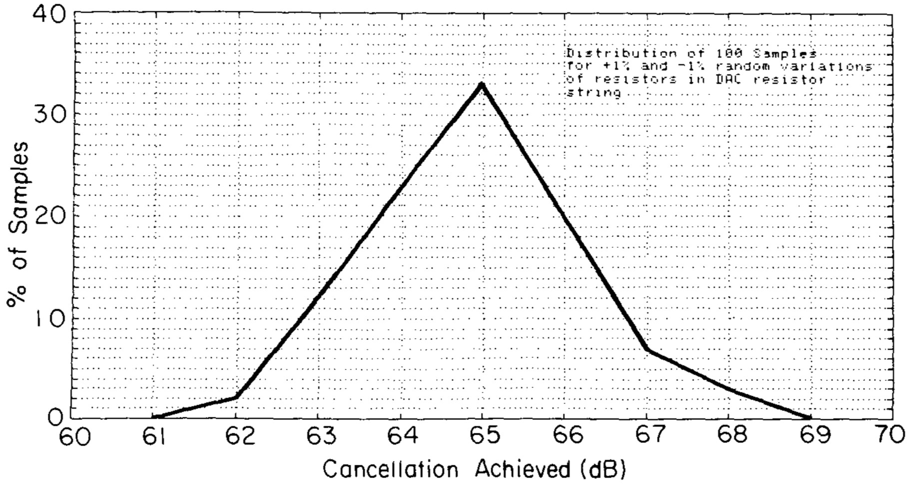



Fig. 13. Histogram showing the results of random +1 and −1 percent perturbations in resistor of the 12 bit DAC of Fig. 10, for 100 samples.

ically smaller than would be required in the table look-up method (1024 for a ten data bit cancellation).

The effect of random photolithographic mismatches in resistors was also simulated. Individual mismatches of either +1 percent or −1 percent (chosen randomly) were added to each of the 16 resistors of a string initially designed to implement a 12 bit D/A with the characteristic of (4.1). This level of mismatch is typical of what would be expected from an MOS process, although it is extremely unlikely that all the resistors would be simultaneously mismatched to this degree. This type of mismatch leads to a continuous piecewise linear characteristic in the D/A. In each simulation the same set of 26 taps shown in Table II were used (although many of them were very small). A histogram of the residual error in 100 randomly chosen mismatches is presented in Fig. 13. There is a considerable spread of 6 dB in the residual error, due to the choice of the same taps in each case (changing which taps were implemented for each random mismatch would presumably narrow this range and improve the cancellation). Table II shows the distribution of the maximum absolute values of the taps. Here a 0 in the intersection of column 0.001 and row $C_0C_1C_2C_3$ means that the corresponding tap was smaller than 0.001 (in absolute value) for all the 100 samples.

TABLE II
MAXIMUM VALUES OF TAPS OVER THE 100 RUNS OF FIG. 13

| TAP NUMBER | PRODUCT TERM | ALWAYS > 0.005 | ALWAYS > 0.001 | ALWAYS > 0.0005 | ALWAYS > 0.0001 | ALWAYS > .00006 | ALWAYS > .00001 |
|---|---|---|---|---|---|---|---|
| 0 | C0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | C1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | C2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | C3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | C4 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | C5 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | C6 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | C7 | 0 | 1 | 1 | 1 | 1 | 1 |
| 8 | C8 | 0 | 0 | 1 | 1 | 1 | 1 |
| 9 | C9 | 0 | 0 | 0 | 1 | 1 | 1 |
| 10 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 11 | C0C1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 12 | C0C2 | 0 | 0 | 1 | 1 | 1 | 1 |
| 13 | C0C3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 14 | C0C4 | 0 | 0 | 0 | 0 | 1 | 1 |
| 15 | C1C2 | 0 | 0 | 1 | 1 | 1 | 1 |
| 16 | C1C3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 17 | C1C4 | 0 | 0 | 0 | 1 | 1 | 1 |
| 18 | C2C3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 19 | C2C4 | 0 | 0 | 0 | 1 | 1 | 1 |
| 20 | C3C4 | 0 | 0 | 0 | 0 | 1 | 1 |
| 21 | C0C1C2 | 0 | 1 | 1 | 1 | 1 | 1 |
| 22 | C0C1C3 | 0 | 0 | 1 | 1 | 1 | 1 |
| 23 | C1C2C3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 24 | C0C2C3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 25 | C0C1C2C3 | 0 | 0 | 0 | 1 | 1 | 1 |

## V. CONCLUSIONS

A technique for compensating for nonlinearities in the echo channel and the echo canceller itself has been proposed. It has the desirable features of requiring a modest increase in canceller complexity for mild nonlinearities and not resulting in a significant slowing of convergence. For a relatively long echo impulse response and mild nonlinearities, it achieves a dramatic reduction of complexity and speedup of convergence relative to the table look-up approach. Initial computer simulations have indicated that the increase in the number of taps is indeed modest for the type of nonlinearities in typical MOS monolithic D/A converters, and that impressive echo attenuations can be obtained using this technique in conjunction with these converters. These results are indeed very encouraging. However, the model of A/D converter nonlinearity would have to be considerably refined before one could state with confidence how many taps would be required to achieve a specified degree of echo cancellation. Factors which would have to be considered in detail would be the processing variations and component mismatches which would be encountered in the manufacture of such a device.

## APPENDIX

The determination of the coefficients of the expansion is easily done as in Section II in terms of the $B_k$, which assume the values 0 and 1. However, it is of interest to transform the expansion into terms of a new variable $C_k$, which assumes two different values. Substituting (2.9) directly into (2.1) results in an inordinate number of cross terms, and is probably intractable. However, the new expansion can be determined as follows. Consider just the terms in (2.1) which contain $B_0$; there are precisely $2^{N-1}$ of them. Factoring $B_0$ out of these terms, we get the decomposition

$$f(B_0, B_1, \cdots, B_{N-1}) = f_1(B_1, \cdots, B_{N-1}) + B_1 f_2(B_1, \cdots, B_{N-1}) \tag{A.1}$$

where $f_1$ and $f_2$ are each expansions identical in form with $2^{N-1}$ terms. Substituting for $B_0$ in terms of $C_0$

$$f(C_0, B_1, \cdots, B_{N-1}) = f_1'(B_1, \cdots, B_{N-1}) + C_0 f_2'(B_1, \cdots, B_{N-1}) \tag{A.2}$$

where

$$f_1'(B_1, \cdots, B_{N-1}) = f_1(B_1, \cdots, B_{N-1}) - \frac{a}{b} f_2(B_1, \cdots, B_{N-1}) \tag{A.3}$$

and

$$f_2'(B_1, \cdots, B_{N-1}) = \frac{1}{a} f_2(B_1, \cdots, B_{N-1}). \tag{A.4}$$

Once all the terms of these two partial expansions have been determined as in (A.3) and (A.4), the two partial expansions can be combined to form a single expansion for $f(C_0, B_1, \cdots, B_{N-1})$. When this procedure is repeated for $B_1$ through $B_{N-1}$, the expansion $f(C_0, C_1, \cdots, C_{N-1})$ is complete.

## REFERENCES


[1] S. V. Ahamed, P. P. Bohn, and N. L. Gottfried, "A tutorial on two-wire digital transmission in the loop plant," *IEEE Trans. Commun.*, vol. COM-29, pp. 1554–1564, Nov. 1981.

[2] M. M. Sondhi and D. A. Berkley, "Silencing echoes on the telephone network," *Proc. IEEE*, vol. 68, pp. 948–963, Aug. 1980.

[3] N. Holte and S. Stueflotten, "A new digital echo canceller for two-wire subscriber lines," *IEEE Trans. Commun.*, vol. COM-29, Nov. 1981.

[4] B. Fotouhi and D. A. Hodges, "High resolution A/D conversion in MOS/LSI," *IEEE J. Solid State Circuits*, vol. SC-14, pp. 920–926, Dec. 1979.

[5] O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Echo canceller for an 80 kb/s baseband modem," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 1982.

[6] K. H. Mueller, "Combining echo cancellation and decision feedback equalization," *Bell Syst. Tech. J.*, vol. 58, pp. 491–500, Feb. 1979.

[7] M. Schetzen, "Nonlinear system modeling based on the Wiener theory," *Proc. IEEE*, vol. 69, p. 1557, Dec. 1981.

[8] K. H. Mueller, "A new digital echo canceller for two-wire full-duplex data transmission," *IEEE Trans. Commun.*, vol. COM-24, pp. 956–962, Sept. 1976.

[9] H. Kobayashi, "A survey of coding schemes for transmission or recording of digital data," *IEEE Trans. Commun.*, vol. COM-19, p. 1087, Dec. 1971.

[10] N. Verhoeckx *et al.*, "Digital echo cancellation for baseband data transmission," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, p. 768, Dec. 1979.

[11] W. Kaplan, *Advanced Calculus.* Reading, MA: Addison-Wesley, 1952.

[12] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE* vol. 57, Jan. 1969.

[13] O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Large scale integration of hybrid-method digital subscriber loops," *IEEE Trans. Commun.*, vol. COM-30, pp. 2095–2108, Sept. 1982.


★

**Oscar Agazzi**, for a photograph and biography, see p. 2108 of the September 1982 issue of this TRANSACTIONS.

★

**David G. Messerschmitt** (S'65–M'68–SM'78), for a photograph and biography, see p. 2108 of the September 1982 issue of this TRANSACTIONS.

★

**David A. Hodges** (S'59–M'65–SM'71–F'77), for a photograph and biography, see p. 2108 of the September 1982 issue of this TRANSACTIONS.