Optimal Decoding for Data Acquisition Applications of Sigma Delta Modulators

Søren Hein, *Member, IEEE*, and Avideh Zakhor, *Member, IEEE*

Abstract—**We propose a class of optimal decoding algorithms for data acquisition applications of sigma delta ($\Sigma\Delta$) modulators. Our technique is applicable to all current $\Sigma\Delta$ structures, including single and double loop, cascade, and interpolative modulators. While the performance of our technique is identical to that of other optimal nonlinear decoding schemes such as table lookup, it is considerably simpler to implement. Numerical results are presented to compare the performance of our decoding technique to that of linear decoders. Effects of circuit imperfections on performance are also examined.**

I. Introduction

SIGMA delta ($\Sigma\Delta$) modulators as analog-to-digital (A/D) converters have received considerable attention from the signal processing community. Their attraction lies in the tradeoff provided between sampling rate and resolution of the in-loop quantizer—specifically, they can achieve the same or higher resolution as multibit quantizers operating at the Nyquist rate by employing a low-resolution quantizer operating at many times the Nyquist rate. In practice, the low-resolution quantizer is usually one-bit because of its ease of implementation and the inherent linearity of the two levels.

$\Sigma\Delta$ modulators generally require fewer and simpler components than comparable converters of different types, and are robust against circuit imperfections. Furthermore, they obviate the need for stringent analog antialiasing filtering, and relegate the strict processing demands to the digital domain. They are thus attractive for VLSI implementation of relatively low-bandwidth signal processing applications, such as speech and audio.

In this paper we investigate the application of $\Sigma\Delta$ modulators to data acquisition. The particular setup we consider for a conversion cycle of the $\Sigma\Delta$ modulator is the following: All initial encoder states are set to zero, and the encoder is run for N cycles with constant input. The resulting N-bit output sequence is fed to a decoder whose task is to estimate the input. N is referred to as the oversampling ratio (OSR) [1]. We emphasize that a different definition of OSR is in use for $\Sigma\Delta$ modulators operating on dynamic inputs [1], and that results in papers on dynamic inputs [2]–[5] are not directly comparable to results in this paper.

In [6], [7] we decoupled a given $\Sigma\Delta$ modulator into its encoder and decoder parts and investigated the encoder separately. The idea was to view the encoder as a source coder or nonuniform quantizer, dividing the dynamic range into intervals separated by transition points, with each interval corresponding to a distinct N-bit output sequence. It was shown that for fixed initial encoder states, only a small fraction of the 2^N possible N-bit sequences can appear at the output, as the constant input is swept over the dynamic range; these sequences will be referred to as codewords. The optimal performance in terms of minimizing the mean squared error (MSE) is achieved by a decoder that takes a codeword as its input, and outputs the midpoint of the corresponding interval.[1] Such a decoder is nonlinear: It exploits the specific bit patterns, rather than a frequency domain representation of them, to arrive at optimal estimates of the input.

The optimal decoder could in principle be implemented using a table in the form of a programmable logic array (PLA). In practice this is not feasible, as the table would be prohibitively large. Here we present a general and simple technique, called zooming, for optimal decoding [8]. We compare the performance of the zoomer to linear decoding under ideal circumstances in Section II, and in the presence of various circuit imperfections in Section III. The encoder structures we consider include the single loop, double loop, two stage noise-shaping (MASH) and interpolative encoders.

II. Optimal Decoding Under Ideal Conditions

This section presents algorithms for optimal nonlinear decoding of the output of various $\Sigma\Delta$ encoders, including the single and double loop, the two stage MASH, and the interpolative encoders. All elements of the encoders are assumed to function ideally; nonidealities are considered in Section III. A summary of the performance results are shown in Table I. Throughout the paper we make the following assumptions:

Manuscript received February 9, 1991; revised November 13, 1992. This work was supported by NSF Grant MIP-8911017, Analog Devices, and the Air Force Office of Scientific Research (AFOSR/JSEP) under Contract F49620-90-C-0029, and ONR Young Investigator Award N00014-92-J-1732. It was also presented in part at the Twenty-Fourth Asilomar Conference on Signals, Systems, and Computers, November 1990.

S. Hein was with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720. He is now with Siemens AG, W-8000 München 83, Germany.

A. Zakhor is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720.

IEEE Log Number 9205107.

[1]This is only strictly true if the random variable X is uniformly distributed on the dynamic range D. It holds in the limit as $N \to \infty$ if X has a smooth probability density function.

TABLE I
COMPARISON OF ZOOMER AND LINEAR DECODING FOR THE IDEAL SINGLE LOOP, DOUBLE LOOP, AND TWO STAGE MODULATORS

	SNR Slope (dB/oct)		WC Slope (bits/oct)		SNR (dB), $N = 128$		WCR (bits), $N = 128$	
	Zoomer	Linear	Zoomer	Linear	Zoomer	Linear	Zoomer	Linear
Single loop	9.0	9.0	1.0	1.0	62.5	54.5	7.8	6.8
Double loop	17.0	14.7	2.3	1.7	92.0	67.0	13.3	8.8
Two stage	18.0	14.7	2.2	2.2	101.0	72.5	13.4	10.7

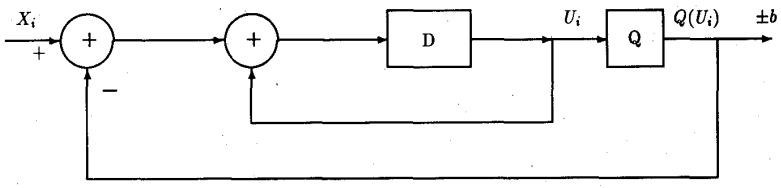


Fig. 1. Discrete-time model of the single loop $\Sigma\Delta$ encoder.

1) One-bit in-loop quantizer, given by

$$Q(U) = \begin{cases} -b & \text{if } U \le 0 \\ +b & \text{if } U > 0 \end{cases} \tag{1}$$

where $B = (-b, +b)$ is the full dynamic range, b is a constant, and U is the quantizer input.[2]

2) Known initial encoder states. In fact, we assume for convenience that these are all initialized to zero before the encoder is started.

3) Constant input. The input X is assumed to be a random variable that takes on constant values in the dynamic range $D \subset B$. The assumption is made because we are focusing on data acquisition applications in which inputs can be assumed more or less constant. In practice, the full dynamic range B is seldom used. One reason is to avoid the possibility of exceeding the dynamic range; another is that the largest estimation errors are generally made when the input is close to $\pm b$ [6], [9]. Therefore we restrict the dynamic range to $D = (-Kb, +Kb)$, where K is chosen throughout to be 0.9.[3]

The performance measures used to compare decoders are the MSE and the worst case (WC) estimation error, or equivalently, the signal-to-noise ratio (SNR) and the WC resolution in bits. These measures are defined in Appendix B-1. For brevity, curves for WC resolution are omitted in this paper, but may be found in [10], and WC simulation results are included in Table I.

A. Single Loop Modulator

The single loop encoder is the simplest $\Sigma\Delta$ encoder. Fig. 1 shows its discrete-time model, consisting of two adders, a delay element D and a one-bit quantizer Q whose function is given by (1). The inner loop is a discrete integrator that operates on the difference between the input and the quantizer output; due to the negative feedback, the encoder seeks to minimize the accumulated difference between input and output. Section II-A1 presents an optimal decoding algorithm under the assumptions stated above, and Section II-A2 presents numerical results.

1) The Zoomer Algorithm: In this section, we present an optimal decoding algorithm under the assumptions of constant input X and known initial integrator state U_0. For reasons detailed below, we refer to this as the zoomer algorithm. The basic idea is to derive a series of bounds on the constant input.

The system shown in Fig. 1 satisfies the following difference equation in terms of the state variable U_i:

$$U_i = U_{i-1} + X_{i-1} - Q(U_{i-1}), \qquad i \ge 1.$$

Assuming that the initial state is $U_0 = 0$, the state at time n is given by

$$U_n = \sum_{i=0}^{n-1} [X_i - Q(U_i)] = \left(\sum_{i=0}^{n-1} X_i\right) - S_n, \qquad n \ge 1 \tag{2}$$

where S_n is the running sum of output bits given by

$$S_n = \sum_{i=0}^{n-1} Q(U_i), \qquad n \ge 1.$$

Assuming constant input, $X_i = X$ for $i \ge 0$, the first sum in (2) equals nX, and for any given codeword, S_n can be found by summation of the known output sequence. With the definition $S_0 \triangleq 0$, we have the recursive relationship

$$S_n = S_{n-1} + Q(U_{n-1}), \qquad n \ge 1.$$

The only information available to the decoder is the N-bit encoder output sequence, $\{Q(U_n), 0 \le n \le N-1\}$, or equivalently, the signs of the quantizer input sequence. Taking (2) into account, this information determines

[2]The presented techniques generalize in an obvious way to multibit quantization.

[3]Setting K to 1 will decrease the average performance of both linear and nonlinear decoders, but the optimal nonlinear decoder will still be superior to the optimal linear decoder.

whether the sum of inputs is greater or less than the sum of outputs at each time n. For each n we can thus derive a bound on the input:

$$X > \bar{X}_n \text{ if } Q(U_n) = +b; \quad X \le \bar{X}_n \quad \text{if } Q(U_n) = -b \tag{3}$$

where $\bar{X}_n$ is the running average given by

$$\bar{X}_n = \frac{1}{n} S_n. \tag{4}$$

The first two bits of any codeword are uninformative, since they are always $Q(U_0) = -b$ and $Q(U_1) = +b$. The reason for this is that $U_0 = 0$ and from (2), $U_1 = X_0 - Q(U_0) = X + b > 0$. But for each $2 \le n \le N - 1$, (3) gives a lower or upper bound on X, for the known quantity $Q(U_n) = +b$ or $-b$, respectively; thus $Q(U_n)$ determines the type of the bound.

Because there are only a finite number of codewords for inputs in the dynamic range, each codeword can be generated by a specific range of inputs; for constant inputs, only approximately $N^2/2$ out of the 2^N possible N-bit sequences, that is, the codewords, can appear at the output [6]. The zoomer is the decoder that uses the succession of lower and upper bounds from (3) to arrive at the sharpest possible lower and upper bounds on the input resulting in a specific codeword. Fig. 2 shows a flowchart of the zoomer algorithm; it consists of an initialization phase, and a loop containing an update of running sums and an update of either the lower or the upper bound. The algorithm uses lower and upper bound registers L and U initialized to the endpoints of the dynamic range. Sweeping n from 2 to $N - 1$, the zoomer maintains the greatest lower bound and the least upper bound in the registers; at each time step, the new bound is compared with previous bounds, and the appropriate bound register updated accordingly. The algorithm extracts all information from the codeword, and results in an optimal decoding procedure. After processing all the N bits from the encoder, the decoder outputs $(L + U)/2$ as its estimate of the input. From the above, any codeword will result in compatible bounds, that is, $L \le U$. Conversely, all noncodewords will result in incompatible bounds; this last fact is shown in Appendix A. The zoomer is mostly linear, but the conditional register updating is nonlinear.

The zoomer approach is reminiscent of successive approximation, but unlike that type of conversion, a new codeword bit is far from certain to produce new information. In fact, the number of codewords with n bits is close to $n^2/2$, and so at time n, there are only approximately $[(n^2 - (n-1))^2]/2 \approx n$ more codewords than at time $n - 1$. This means that for most of the $n^2/2$ codewords, the last bit is uniquely determined by the previous bits, and for these codewords it carries no information at all.

The calculations involved in the algorithm are quite simple, the division in (4) being the most time-consuming. However, it is not necessary to actually do the divisions until the end; for example, to decide whether or not $X_n > X_m$, we only need to check whether

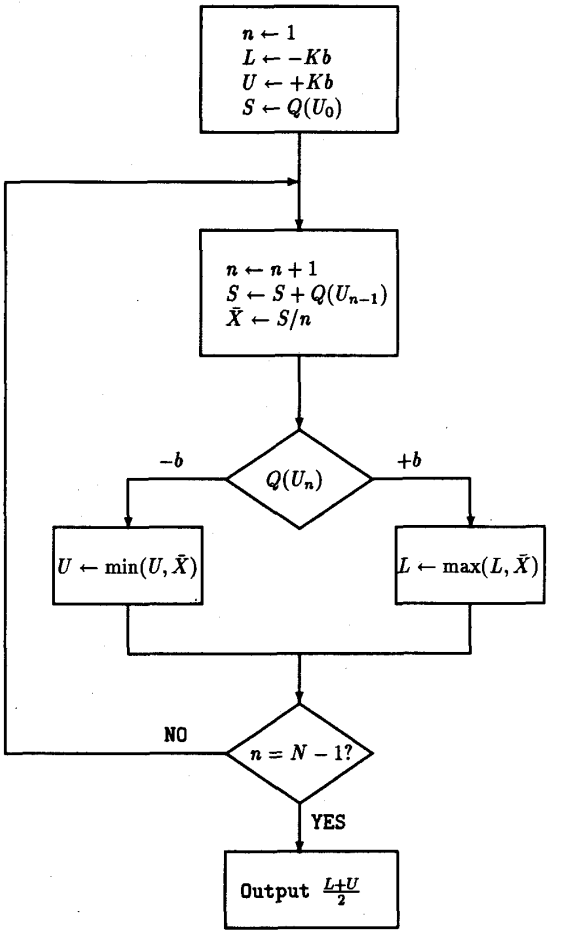


Fig. 2. Flowchart for the single loop zoomer algorithm.

$$mS_n > nS_m. \tag{5}$$

This reduces to integer multiplications rather than floating-point divisions. Instead of storing the best lower and upper $\bar{X}_n$ in the bound registers L and U, we then need to store the best pairs (n, S_n).

Along the same lines, consider the case where p negative bits are interspersed between two positive bits:

$$\{Q(U_n), Q(U_{n+1}), \cdots, Q(U_{n+p}), Q(U_{n+p+1})\} = \{+b, -b, -b, \cdots, -b, -b, +b\}. \tag{6}$$

We wish to compare the lower bounds on X resulting at time n and $n + p + 1$. It can be shown that the difference between the running averages at these times is given by

$$\bar{X}_{n+p+1} - \bar{X}_n = \frac{b + \bar{X}_n}{n + p + 1}\left[\frac{b - \bar{X}_n}{b + \bar{X}_n} - p\right].$$

Since the factor in front of the square brackets is positive, the new lower bound at time $n + p + 1$ is at least as sharp as the old bound at time n if and only if the term in the

square bracket is also positive, that is

$$p \leq \frac{b - \overline{X}_n}{b + \overline{X}_n}$$

or equivalently

$$\overline{X}_n \leq -\frac{p-1}{p+1} b \tag{7}$$

or equivalently

$$(p+1)S_n \leq -n(p-1)b. \tag{8}$$

These expressions can be used in two different ways. First, note that for $p = 0$ the inequality for $\overline{X}_n$ in (7) is always satisfied. This means that if two adjacent bits are both positive, only the bound corresponding to the second bit needs to be calculated, since the bound due to the first bit is guaranteed to be inferior to that of the second one. The decoder can easily check for this by looking one bit ahead in the codeword.[4]

We can also use (8) to compare two lower bounds for more general p. As a replacement for (5), this is only useful if the bound $\overline{X}_n$ at time n is the best lower bound at time n. If so, the bound at time $n + p + 1$ is at least as sharp at that at time n if and only if (8) holds. Similar derivations can be used to compare two upper bounds for the case where p positive bits are interspersed between two negative bits.

To assess the value of using (8) rather than (5), the work involved must be compared. The comparison in (8) calls for integer multiplications to replace (5) which also contains integer multiplications. However, the integer factors in (8) are simple, especially for small p. For instance, all that needs to be checked for $p = 1$ is the sign of S_n. For $p = 2$, the factor $(p - 1)$ reduces to 1, and for $p = 3$, both $(p - 1)$ and $(p + 1)$ are powers of 2. Since small values of p are more likely to occur, this could potentially save computation.

2) Numerical Results: We now present numerical results on the performance of the zoomer and compare it to the asymptotically optimal linear N-tap finite impulse response (FIR) decoder for constant inputs. This filter was derived by Gray [11] and has tap coefficients

$$h_n = 6\frac{(n+1)(N-n)}{N(N+1)(N+2)}, \qquad 0 \leq n \leq N-1. \tag{9}$$

Fig. 3 shows that for a given oversampling ratio, the zoomer gains about 8 dB or $1\frac{1}{2}$ bits of SNR over the FIR filter. Alternatively, the zoomer requires half the oversampling ratio of the FIR filter to obtain essentially the same performance, resulting in shorter data acquisition times. Both SNR curves have a slope of 9 dB/octave.

[4]Look-ahead techniques are routinely used in implementation of signal processing algorithms.

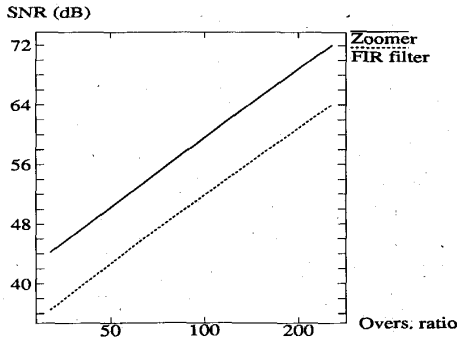


Fig. 3. Single loop encoder: SNR as a function of oversampling ratio for the zoomer and the asymptotically optimal FIR filter.

A direct comparison of these results to those obtainable with other types of A/D converters is difficult, but for purposes of illustration we consider the dual slope converter. We assume the required number of clock cycles to be comparable to the oversampling ratio of a $\Sigma\Delta$ modulator. A dual slope converter using P clock cycles for full-scale inputs has the effect of dividing the dynamic range into $P/2$ intervals of width $4b/P$ each, leading to an MSE of $4b^2/3P^2$ and an SNR of $20 \log_{10} P$. To match the SNR of the single loop zoomer at an oversampling ratio of 128, the dual slope converter thus requires approximately 3000 cycles.

B. Double Loop Modulator

The double loop encoder is a generalization of the single loop encoder that has a more favorable tradeoff between resolution and oversampling ratio [4]; its discrete-time model is shown in Fig. 4. The encoder is important in its own right, and as a building block is cascaded modulators. The encoder contains two cascaded discrete integrators, and the quantizer output is fed back to the input as well as to an intermediate node. Section II-B1 presents an optimal decoding scheme under the assumptions stated in the beginning of Section II, and Section II-B2 presents simulation results.

1) The Zoomer Algorithm: The analysis of the double loop encoder proceeds in a fashion similar to that of the single loop encoder. The governing difference equations for the system shown in Fig. 4 are

$$U_n = U_{n-1} + V_{n-1} - Q(U_{n-1})$$

$$V_n = V_{n-1} + X_n - Q(U_n), \qquad n \geq 1$$

where U_n and V_n are the two integrator state variables. Assuming zero initial states, $U_0 = V_0 = 0$, and constant input, $X_n = X$, the equation for V_n implies

$$\sum_{i=0}^{n-1} V_i = \tfrac{1}{2} n(n-1)X - \sum_{i=1}^{n-1} (n-i)Q(U_i)$$

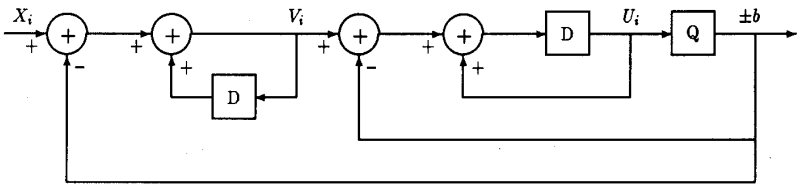


Fig. 4. Discrete-time model of the double loop $\Sigma\Delta$ encoder.

and the equation for U_n implies

$$U_n = \sum_{i=0}^{n-1} V_i - \sum_{i=0}^{n-1} Q(U_i)$$
$$= \tfrac{1}{2} n(n-1)X - Q(U_0)$$
$$- \sum_{i=1}^{n-1} (n-i+1)Q(U_i), \qquad n \geq 2. \quad (10)$$

To facilitate the calculation of the sum involving quantizer outputs, we define the running sums

$$S_n = \sum_{i=1}^{n-1} Q(U_i), \qquad n \geq 2 \quad (11)$$

$$W_n = \sum_{i=1}^{n-1} (n-i+1)Q(U_i), \qquad n \geq 2. \quad (12)$$

Defining $S_1 = W_1 \triangleq 0$, we then wave the recursions

$$S_n = S_{n-1} + Q(U_{n-1}), \qquad n \geq 2 \quad (13)$$

$$W_n = W_{n-1} + S_n + Q(U_{n-1}), \qquad n \geq 2. \quad (14)$$

The quantity W_n can thus be found as a weighted summation of the output sequence, as shown in (12).

The information available to the decoder is $\{Q(U_n), 0 \leq n \leq N-1\}$. From the difference equations we have $U_1 = +b > 0$, $V_1 = X - b$ and $U_2 = X - b < 0$, so the first three output bits are always $Q(U_0) = -b$, $Q(U_1) = +b$ and $Q(U_2) = -b$ regardless of the input, and the first informative bit is $Q(U_3)$. As in the single loop case, we can use (10) to obtain a bound on the input at each time $3 \leq n \leq N-1$: we obtain a lower or an upper bound depending on whether $Q(U_n) = +b$ or $-b$. Specifically,

$$X > \overline{X}_n \qquad \text{if } Q(U_n) = +b;$$
$$X \leq \overline{X}_n \qquad \text{if } Q(U_n) = -b$$

where $\overline{X}_n$ is given by

$$\overline{X}_n = \frac{Q(U_0) + W_n}{\frac{1}{2}n(n-1)}, \qquad n \geq 3. \quad (15)$$

Recall that each codeword is generated by a specific range of input values. Analogous to the single loop zoomer, the double loop zoomer is the decoder that uses the output sequence to derive a succession of lower and upper bounds on the input; the sharpest of these bounds are the best possible bounds on the input resulting in the output sequence. This is achieved by using the lower and upper bound registers L and U, initialized to the endpoints of the dynamic range. Sweeping n from 3 to $N-1$, the zoomer maintains the greatest lower bound and the least upper bound in the registers. To be specific, Fig. 5 show a flowchart of the double loop zoomer algorithm. The variables S and W correspond to the quantities given by (13) and (14), respectively, and p is the denominator in the bound fraction (15).

During the conversion cycle, the floating-point division in (15) can be replaced by integer multiplication, since comparisons of the form $\overline{X}_n > \overline{X}_m$ can be written in the simpler form

$$m(m-1)[Q(U_0) + W_n] > n(n-1)[Q(U_0) + W_m].$$

As in the single loop case, there exist alternative formulas to decide whether or not a new bound is better than an old bound in the special case described by (6). However, the formulas are more involved for the double loop than for the single loop encoder. Even when two successive bits are identical, there is no guarantee that the second bound is tighter than the first one.

2) Numerical Results: This section compares the performance of the double loop zoomer to that of linear decoding. There is no parallel in the literature to the asymptotically optimal FIR filter (9) for single loop modulation. The linear decoder under consideration here is therefore chosen to be the N-tap filter with a sinc^3 transfer function which is believed to be close to optimal [5].

Fig. 6 shows that at a given oversampling ratio, the zoomer is superior to the sinc^3 filter by between 20 and 30 dB of SNR. The SNR achieved by the sinc^3 filter at an oversampling ratio of 256 is reached by the zoomer at an OSR of approximately 100. This translates into shorter data acquisition times. For the zoomer, the slope of the SNR curves is about 17 dB/octave, whereas for the linear filter, the slope is 14.7 dB/octave. It is thus seen that ideally, the zoomer achieves a better trade-off with oversampling ratio than the linear filter, and the gap between the curves widens as the oversampling ratio increases.

C. Two Stage Modulator

Fig. 7 shows the discrete-time model of the two stage MASH encoder. The MASH architecture was originally proposed by Uchimura *et al.* [13] and has been extensively analyzed by Wong, Chou, and Gray in several papers, including [12] and [14]. The encoder consists of two single loop stages, of which the first is fed with the input, and the second is fed with the quantization error sequence of the first stage.

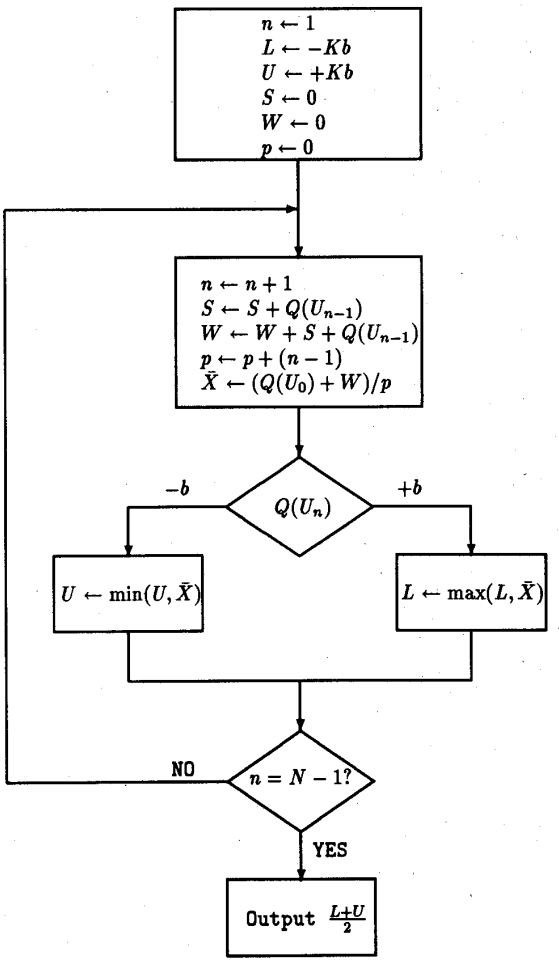


Fig. 5. Flowchart for the double loop zoomer algorithm.

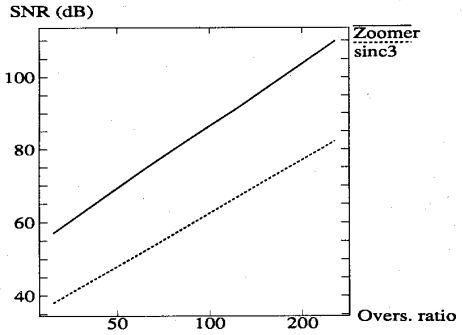


Fig. 6. Double loop encoder: SNR as a function of oversampling ratio for the zoomer and the sinc^3 filter.

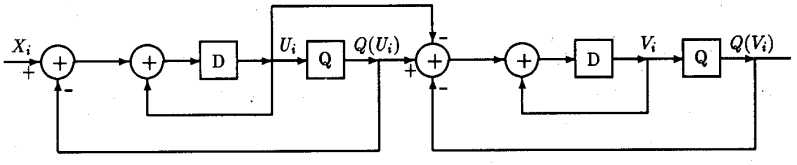


Fig. 7. Discrete-time model of the two stage $\Sigma\Delta$ encoder.

The original papers on this cascade structure also include a noise canceling circuit that performs noise shaping and combines the two binary output streams into one quaternary sequence [13], [12]. This has the effect of eliminating the direct appearance of the first stage quantization error in the output sequence. It should be noted that although this is a desirable characteristic, the circuit might in general be discarding information present in the separate stage outputs. In addition, for data acquisition applications, the noise shaping characteristic is of little importance since the input is more or less constant. Here we will adopt the viewpoint that the noise cancelling circuit is part of a decoder, and the decoder should not be limited to operating on the sequence obtained by irreversibly combining the two output sequences into one. We will therefore use $\{Q(U_i)\}$ and $\{Q(V_i)\}$ directly for decoding. Section II-C1 presents an optimal decoding scheme under the assumptions stated in Section II, and Section II-C2 presents simulation results.

1) The Zoomer Algorithm: The difference equations governing the state variables U_n and V_n for the two stage encoder shown in Fig. 7 are

$$U_n = U_{n-1} + X_{n-1} - Q(U_{n-1})$$

$$V_n = V_{n-1} - U_{n-1} + Q(U_{n-1}) - Q(V_{n-1}), \qquad n \geq 1.$$

Assuming that the initial states are $U_0 = V_0 = 0$, these can be solved to yield

$$U_n = \sum_{i=0}^{n-1} X_i - \sum_{i=0}^{n-1} Q(U_i), \qquad n \geq 1 \tag{16}$$

$$V_n = -\sum_{i=0}^{n-2} (n-1-i)X_i + \sum_{i=0}^{n-1} (n-i)Q(U_i) - \sum_{i=0}^{n-1} Q(V_i), \qquad n \geq 2. \tag{17}$$

Let us define the running sums

$$S_n = \sum_{i=0}^{n-1} Q(U_i), \qquad n \geq 1 \tag{18}$$

$$T_n = \sum_{i=0}^{n-1} Q(V_i), \qquad n \geq 1 \tag{19}$$

$$W_n = \sum_{i=0}^{n-1} (n-i)Q(U_i), \qquad n \geq 1. \tag{20}$$

Defining $S_0 = T_0 = W_0 \triangleq 0$ we then have the recursions

$$S_n = S_{n-1} + Q(U_{n-1}), \qquad n \geq 1$$

$$T_n = T_{n-1} + Q(V_{n-1}), \qquad n \geq 1$$

$$W_n = W_{n-1} + S_n, \qquad n \geq 1.$$

As before we assume that the input is constant, $X_i = X$ for $i \geq 0$. At time n, (16) and (17) each provide potential new bounds on this input: The new bound is an upper or a lower bound depending on whether $Q(U_n) = -b$ or $+b$. Specifically, (16) results in

$$X > \overline{X}_n^{(1)} \quad \text{if } Q(U_n) = +b;$$

$$X \leq \overline{X}_n^{(1)} \quad \text{if } Q(U_n) = -b; \qquad n \geq 1 \tag{21}$$

where $\overline{X}_n^{(1)}$ is the running average

$$\overline{X}_n^{(1)} = \frac{1}{n} S_n. \tag{22}$$

Equation (17) results in an upper or lower bound on the input, depending on whether $Q(V_n)$ is positive or negative. Specifically,

$$X < \overline{X}_n^{(2)} \quad \text{if } Q(V_n) = +b;$$

$$X \geq \overline{X}_n^{(2)} \quad \text{if } Q(V_n) = -b; \qquad n \geq 2 \tag{23}$$

where

$$\overline{X}_n^{(2)} = \frac{W_n - T_n}{\frac{1}{2}n(n-1)}. \tag{24}$$

Fig. 8 shows a flowchart for the two stage zoomer algorithm; the two stage zoomer uses the succession of lower and upper bounds from both (21) and (23) to arrive at overall lower and upper bounds on the input resulting in a specific codeword. This is achieved by using lower and upper bound registers L and U, initialized to the endpoints of the dynamic range. Sweeping n from 3 to $N-1$, the zoomer maintains the greatest lower bound and the least upper bound in the registers. Variables S, T, and W in Fig. 8 hold the sums in (18)–(20), respectively, and p is the denominator of (24), while $\overline{X}^{(1)}$ and $\overline{X}^{(2)}$ correspond to the quantities (22) and (24). At each time step, the flowchart contains two conditional updates of the bound registers, corresponding to (22) and (24).

2) Numerical Results: This section compares the performance of the two stage zoomer to an N-tap filter with a sinc^3 transfer function. It is shown in [12] that this filter achieves an MSE of $O(N^{-5})$, and that no sinc^k filter can achieve a better asymptotic dependence on N.

Fig. 9 shows that at a given oversampling ratio, the zoomer outperforms the sinc^3 filter by 20–30 dB of SNR. For the depicted range of oversampling ratios, this translates into a reduction by a factor of 2–3 in data acquisition times to achieve a given performance. For the zoomer, the slope of the SNR curve is about 18 dB/octave, whereas for the linear filter, the slope is 14.7 dB/octave. It is thus seen that ideally, the zoomer achieves a more favorable tradeoff between SNR and oversampling ratio than the linear filter, and the gap between the curves widens as the oversampling ratio increases: The MSE goes as $O(N^{-6})$ and $O(N^{-5})$, respectively. Compared to the double loop results in Fig. 6, the slope difference for the two stage encoder is about 1 dB/octave greater for the SNR curves.

D. Interpolative Modulators

The general interpolative encoder structure is shown in Fig. 10 [9]. It is characterized by the transfer function

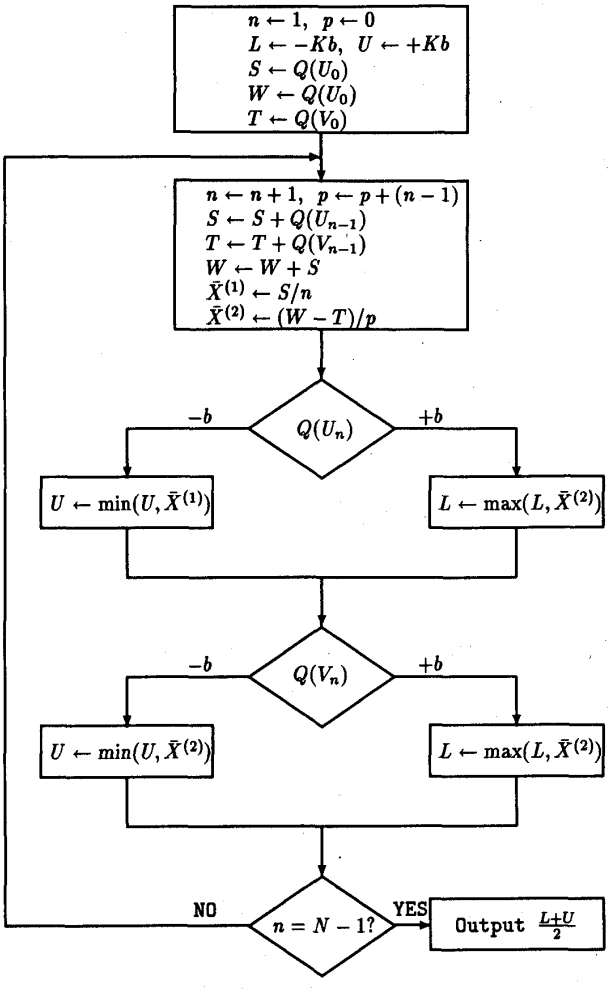


Fig. 8. Flowchart for the two stage zoomer algorithm.

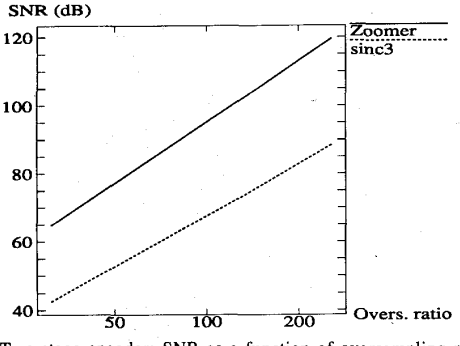


Fig. 9. Two stage encoder: SNR as a function of oversampling ratio for the zoomer and the sinc^3 filter.

$H(z)$ which is chosen to be low pass. This section demonstrates the applicability of the zoomer concept to interpolative encoders.

Let us denote the impulse response of $H(z)$ by $\{h_0, h_1, h_2, \cdots\}$. To avoid race-around we must have $h_0 = 0$. Under the constant input assumption, the state variable U

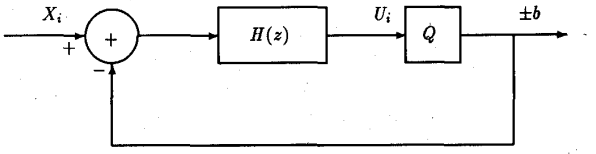


Fig. 10. Discrete-time model of the interpolative $\Sigma\Delta$ encoder.

at time n is given by

$$U_n = \sum_{i=0}^{n} h_{n-i}[X_i - Q(U_i)]$$

$$= X\left(\sum_{i=0}^{n-1} h_{n-1}\right) - \sum_{i=0}^{n-1} h_{n-i}Q(U_i), \qquad n \geq 0.$$

The zoomer algorithm uses the output bits to derive a succession of upper and lower bounds on the input. The bound at time n is a lower or an upper bound for $Q(U_n) = +b$ and $-b$, respectively. Specifically,

$$X > \bar{X}_n \quad \text{if } Q(U_n) = +b;$$

$$X \leq \bar{X}_n \quad \text{if } Q(U_n) = -b$$

where $\bar{X}_n$ is given by

$$\bar{X}_n = \frac{\sum_{i=0}^{n-1} h_{n-i}Q(U_i)}{\sum_{i=1}^{n} h_n}.$$

As an example, the single loop encoder can be viewed as an interpolative encoder with $H(z) = z^{-1}/(1 - z^{-1})$. Therefore $\{h_0, h_1, h_2, \cdots\} = \{0, 1, 1, \cdots\}$ and

$$\bar{X}_n = \frac{1}{n}\sum_{i=0}^{n-1} Q(U_i)$$

in agreement with (4). Note that neither the double loop nor the two stage encoder are interpolative encoders.

The above technique readily generalizes to the case where the transfer function from input to quantizer input is different from the transfer function from quantizer output to quantizer input. This would be the case if there was also a filter in the feedback path from quantizer input to input summing node; see for example [15].

III. Decoding in the Presence of Nonidealities

In this section we use simulations to investigate the effects of nonidealities on the performance of the zoomer algorithms for the single and double loop encoders and the two stage encoder. To realistically assess the performance of our nonlinear decoder, we consider a variety of circuit imperfections. A summary of the results of this section is shown in Table II. As our results are based on simulation, they must be considered tentative; practical circuit implementations are needed to verify them experimentally.

For each encoder, the following types of nonidealities have been investigated: Nonunity integrator gain, integra-

TABLE II
SENSITIVITY OF THE ZOOMER TO VARIOUS NONIDEALITIES. QUANTITIES GIVEN IN PERCENT ARE MEASURED RELATIVE TO THE QUANTIZER STEP SIZE b. QUANTITIES GIVEN FOR THE LEAK ARE THE MINIMUM OP-AMP GAINS NECESSARY

	Nonideality to cause 2-dB SNR loss at $N = 128$		
	Single Loop	Double Loop	Two Stage
Gain	N/A	0.1%	0.02%
Leak	2 500	50,000	100,000
Initial state	1.2%	0.05%	0.01%
DC offset	1.2%	0.5%	0.01%
Noise	0.5%	0.02%	0.005%
Input var.	0.2%	0.01%	0.005%

tor leak, nonzero initial integrator state, DC quantizer offset and random noise. Appendix C quantifies the form the nonidealities are assumed to take. While it is recognized that practical encoders suffer to some degree from all of these nonidealities, their effects are considered separately here to facilitate assessments. For each combination of encoder structure and nonideality, we have derived a modification to the zoomer algorithm that can be used if the numerical value of the nonideality is known in advance. To distinguish between the modified zoomers and the ones described in Section II, the latter are referred to as original zoomers when necessary. For brevity, we only include here the derivations for the double loop and two stage encoders; the single loop encoder is comparatively less important due to its inferior tradeoff between oversampling ratio and SNR. The derivations for the double loop encoder are shown in Section III-A, and the derivations for the two stage encoder can be found in Appendix D. We omit the treatment of sloped inputs which leads to a two-dimensional linear programming problem [10]. Detailed simulation results can be found in [10], and are summarized in Table II.

With the modifications of the original zoomers which we derive, the concepts of transition points and quantization intervals are still well defined; each nonideal encoder can be specified by its quantization intervals, transition points, and codewords. However, using an original zoomer to decode an output sequence of a nonideal encoder will not in general result in an estimate which is the midpoint of the actual quantization interval, as specified by the characteristics of the nonideal encoder. This is because the quantization intervals of a nonideal encoder change as functions of the numerical values of the nonidealities. In fact, an original zoomer estimate based on an output sequence from a nonideal encoder may be outside the actual quantization interval, and the original zoomer bounds L and U may be incompatible.[5] Using an original zoomer thus results in performance degradation in the presence of nonidealities. In Appendix B-2 we define the performance measures of SNR and worst case resolution in the presence of nonidealities. We now show simulation results to quantify these degradations when the nonidealities must be considered unknown.

[5]In the case of incompatible bounds, the original zoomer estimate is still defined as the average $(L + U)/2$.

A. Double Loop Modulator

This section considers the effects of nonidealities on the double loop zoomer algorithm. Throughout we compare the zoomer to the linear N-tap decoder with a sinc^3 transfer function.

1) Nonunity Integrator Gains: Denoting the gains of the first and second integrator by g_1 and g_2, respectively, the difference equations for the double loop encoder can be written

$$U_n = U_{n-1} + g_2(V_{n-1} - Q(U_{n-1}))$$
$$V_n = V_{n-1} + g_1(X_n - Q(U_n)), \qquad n \geq 1$$

and (10) becomes

$$U_n = g_2\left(g_1 \sum_{i=1}^{n-1} (n-i)X_i - Q(U_0) - g_1 \sum_{i=1}^{n-1} (n-i+1)Q(U_i) - (1-g_1) \sum_{i=1}^{n-1} (n-i)Q(U_i)\right).$$

It is seen that the gain of the second integrator has no effect on the sign of U_n, and hence it does not affect the performance of any decoder. The zoomer algorithm in the presence of non-unity gain remains unchanged, except that (15) becomes

$$\bar{X}_n = \frac{Q(U_0) + g_1 W_n + (1-g_1)S_n}{\frac{1}{2}g_1 n(n-1)} \tag{25}$$

with S_n and W_n given by (11) and (12). If the gain is known, it can thus be compensated for; in this case, performance is essentially identical to that of the original zoomer described in Section II-B1.

If the gain is unknown, it can be adaptively estimated from (25) in the following way: Initially, we assume g_1 to be 1. Each time a noncodeword appears due to nonunity gain, the lower and upper zoomer bounds will be inconsistent; let us denote these bounds by $\bar{X}_n$ and $\bar{X}_m$, respectively. We can then choose g_1 such that the bounds at time instants n and m become identical; that is, choose g_1 as close as possible to its nominal value of 1 while keeping the bounds barely consistent. To be specific, g_1 is chosen according to

$$g_1 = \frac{m(m-1)(Q(U_0) + S_n) - n(n-1)(Q(U_0) + S_m)}{n(n-1)(W_m - S_m) - m(m-1)(W_n - S_n)}. \tag{26}$$

The estimate of X in the particular cycle in which g_1 is updated will not be so accurate, but the accuracy of subsequent estimates will improve. Once g_1 is well esti-

mated, the performance is essentially identical to that of the original zoomer with an ideal decoder.

We now assume that the integrator gains are unknown, and that the adaptive estimation scheme of (26) is not used. Fig. 11 shows SNR curves for the original zoomer when the first integrator gain is nonunity. It is seen that at an oversampling ratio of 127, an SNR degradation of 2 dB results for a gain which deviates about 0.1% from its nominal value.

2) Leaky Integrators: We make the assumption that the two integrators have the same leak factor α, that is, the op-amps have the same gain. The difference equations then become

$$U_n = \alpha U_{n-1} + V_{n-1} - Q(U_{n-1})$$

$$V_n = \alpha V_{n-1} + X_n - Q(U_n), \qquad n \geq 1.$$

Let us define

$$S_n = \sum_{i=1}^{n-1} \alpha^{n-i-1} Q(U_i), \qquad n \geq 2$$

$$W_n = \sum_{i=1}^{n-1} \alpha^{n-i-1}(n-i+1) Q(U_i), \qquad n \geq 2.$$

Note that if we define $S_1 = W_1 \triangleq 0$,

$$S_n = \alpha S_{n-1} + Q(U_{n-1}), \qquad n \geq 2$$

$$W_n = \alpha W_{n-1} + S_n + Q(U_{n-1}), \qquad n \geq 2.$$

These recursive relations mimic the effect of the two integrators. With leaky integrators, (10) changes into

$$U_n = \frac{(n-1)\alpha^n - n\alpha^{n-1} + 1}{(1-\alpha)^2} X - \alpha^{n-1} Q(U_0) - W_n$$

and the zoomer bound (15) becomes

$$\bar{X}_n = \frac{(1-\alpha)^2}{(n-1)\alpha^n - n\alpha^{n-1} + 1} [\alpha^{n-1} Q(U_0) + W_n]. \tag{27}$$

From the above discussion it is seen that if the integrator leaks are known, they can be compensated for, and in this case, the performance is essentially identical to that of the original zoomer considered in Section II-B2.

We now assume that the integrator leaks are unknown. Fig. 12 shows performance curves for the original zoomer in the presence of leak in both integrators. It is seen that op-amp gains on the order of 50 000 result in an SNR loss of 2 dB.

It is commonly held that nonidealities near the input summing node have relatively more impact on performance than nonidealities further inside the loop. To quantify this, we will show simulations below of the effect of leak in the second integrator, assuming that the first integrator is ideal. In this case, the difference equations are

$$U_n = \alpha U_{n-1} + V_{n-1} - Q(U_{n-1})$$

$$V_n = V_{n-1} + X_n - Q(U_n), \qquad n \geq 1.$$

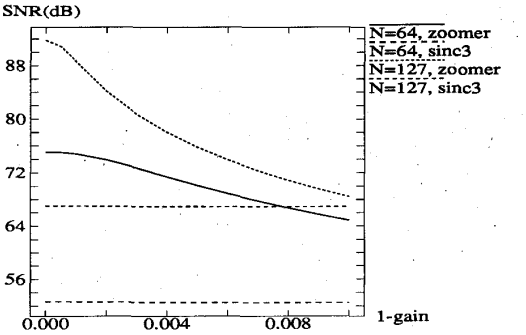


Fig. 11. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of the first integrator gain g_1.

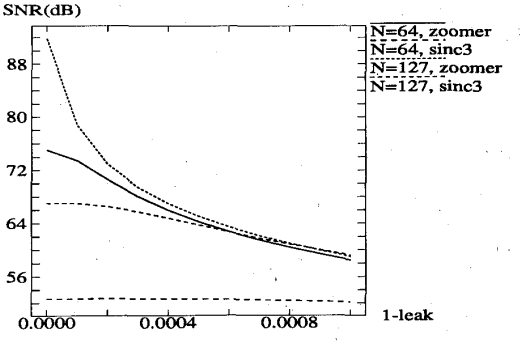


Fig. 12. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of the integrator leak α. Same leak is assumed for both integrators.

We can define the running sums

$$S_n = \sum_{i=1}^{n-1} \alpha^{n-i-1} Q(U_i), \qquad n \geq 2$$

$$W_n = \sum_{i=1}^{n-1} \frac{1-(2\alpha-1)\alpha^{n-i-1}}{1-\alpha} Q(U_i), \qquad n \geq 2.$$

If we define $S_1 = W_1 \triangleq 0$, we have the recursions

$$S_n = \alpha S_{n-1} + Q(U_{n-1}), \qquad n \geq 2$$

$$W_n = \alpha S_{n-1} + S_n + Q(U_{n-1}), \qquad n \geq 2.$$

The zoomer bound becomes

$$\bar{X}_n = \frac{(1-\alpha)^2}{\alpha^n - n\alpha + (n-1)} [\alpha^{n-1} Q(U_0) + W_n].$$

This can be compared with (27), in which both the integrator leaks are assumed to equal α.

Fig. 13 shows the isolated effect of the second integrator leak. Comparing with Fig. 12, it is observed that the sensitivity to the second integrator leak is on the order of 10 times smaller than that to the first integrator leak; therefore the degradation in Fig. 12 is dominated by the first leak.

3) Nonzero Initial States: We make the assumption that the initial state offsets are the same for both state vari-

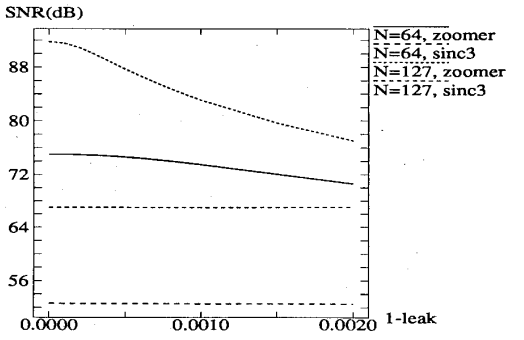


Fig. 13. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of the second integrator leak α. First integrator is assumed ideal.

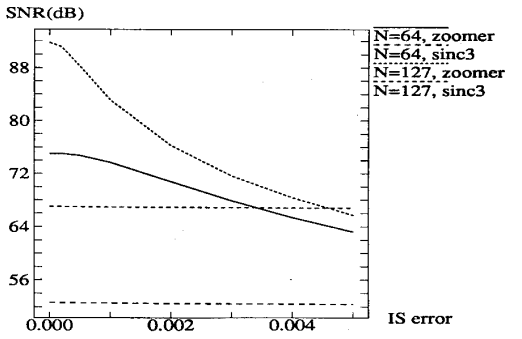


Fig. 14. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of the initial state offset δ. Same offset is assumed for both integrators. The offset is normalized by b.

ables, $U_0 = V_0 = \delta$. It can be shown that the state U at time n then can be written as

$$U_n = U_0 + nV_0 + \sum_{i=0}^{n-1} (n - i)X_i - Q(U_0) - \sum_{i=1}^{n-1} (n - i + 1)Q(U_i).$$

The zoomer bound (15) therefore changes into

$$\bar{X}_n = \frac{-(n + 1)\delta + Q(U_0) + W_n}{\frac{1}{2}n(n - 1)} \tag{28}$$

with W_n given by (12). Note that V_0 contributes $n\delta$, while U_0 only contributes δ. This indicates that more attention should be paid to the initial sate of the first integrator. This is in line with the general belief mentioned above that nonidealities closer to the input summing node have more impact on performance than nonidealities further inside the loop. A qualitative argument in support of this observation is that nonidealities further inside the loop are better suppressed by the negative feedback.

If the initial integrator states are known, they can be compensated for as shown in (28). We now assume that these states are unknown; Fig. 14 shows their effect. It is seen that for an oversampling ratio of 127, an SNR loss of 2 dB results when δ equals about 0.05% of the dynamic range.

4) DC Offset in Quantizer: If the 1-b quantizer is characterized by (37), that is, by the offset q_0, the zoomer bound (15) changes into

$$\bar{X}_n = \frac{q_0 + Q(U_0) + W_n}{\frac{1}{2}n(n - 1)} \tag{29}$$

with W_n given by (12). Thus, a DC offset of q_0 is equivalent to a nonzero initial state of $U_0 = -q_0$. If the DC offset is known, it can be compensated for using (29).

We now assume that the DC offset is unknown; Fig. 15 shows its effect on performance of the original zoomer. Since DC offset and nonzero initial state U_0 are equivalent, and nonzero initial state V_0 is known from Section III-A3 to degrade performance more seriously than non-

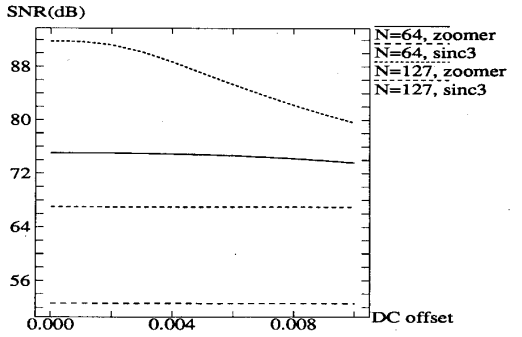


Fig. 15. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of quantizer DC offset q_0. Same offset is assumed for both integrators. The offset is normalized by b.

zero U_0, we expect the sensitivity towards DC offset to be less than that towards initial states shown in Fig. 14. This is confirmed by Fig. 15. For an oversampling ratio of 127, an offset of 0.5% results in an SNR loss of about 2 dB.

5) Random Noise: Consider noisy inputs of the form (39) specified in Appendix C. If the noise sequence were known, the modified zoomer algorithm would involve replacing (15) with

$$\bar{X}_n = -\frac{1}{\frac{1}{2}n(n - 1)} \sum_{i=1}^{n-1} (n - i)N_i + \frac{Q(U_0) + W_n}{\frac{1}{2}n(n - 1)}.$$

The variance of the right-hand side is

$$\frac{4}{n^2(n - 1)^2} \sum_{i=1}^{n-1} \frac{(Mb)^2}{3} i^2 = \frac{2(2n - 1)(Mb)^2}{9n(n - 1)}.$$

Since the noise sequence is unknown in practice, one possible countermeasure against it is to loosen the bound in (15) by two standard deviations.[6] This changes (15) into

$$\bar{X}_n = \frac{Q(U_0) + W_n}{\frac{1}{2}n(n - 1)} - 2\sqrt{\frac{2(2n - 1)}{n(n - 1)}} \frac{MQ(U_n)}{3}. \tag{30}$$

[6]Simulations indicate that it is better to use two standard deviations than one, but no claim is made that two is optimal.

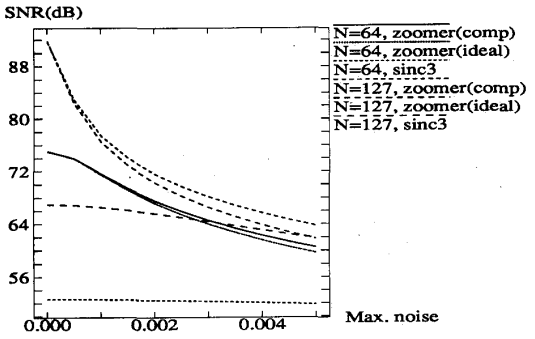


Fig. 16. Double loop encoder: SNR for oversampling ratios 64 and 127 as a function of the maximum normalized noise level M. Curves are shown for the original zoomer, and the compensated zoomer which uses (30).

Note that the sign of the noise correction is opposite that of $Q(U_n)$, so that the bound is loosened. Knowledge of the maximum noise level Mb is required.

Fig. 16 shows the effect of random noise on the input. At an oversampling ratio of 127, an SNR loss of 2 dB results for a maximum noise level of about 0.02%.

IV. Conclusions

We have introduced a general technique for optimal decoding of the output of ideal $\Sigma\Delta$ encoders, under the assumptions of constant input and known initial integrator states. The technique is based on deriving a succession of upper and lower bounds on the input interval generating a given output sequence. The optimal decoder is nonlinear, as might be expected from the nonlinear nature of the encoder. Our results indicate that under ideal circumstances, substantial improvements in SNR and worst case error can be achieved; these results are summarized in Table I. The improvements can be exploited as enhanced performance at the same data acquisition time, or alternatively, as substantial reductions in data acquisition time for the same performance. We have also presented simulation results for performance in the presence of various circuit imperfections; these results are summarized in Table II. It is interesting to compare the sensitivities of the zoomers for the encoder structures considered. In general, the single loop zoomer is the least sensitive at a given OSR. The double loop and two-stage zoomers both achieve better tradeoffs with oversampling ratio than does the single loop zoomer, but they are also more sensitive to non-idealities. The ideal performance of the double loop and two-stage zoomers is roughly comparable, but it is seen that in general, the two-stage zoomer is about 2 to 5 times more sensitive towards circuit imperfections. Future work will be directed towards implementing a working prototype of the zoomer algorithm.

Appendix A

We will show by contradiction that all noncodewords result in incompatible bounds. Assume that a noncodeword gives rise to zoomer bounds $L' < U'$. Choose any input satisfying $L' < X' < U'$, and determine the corresponding codeword. Since the codeword and the noncodeword are different, we can find the earliest time step n_0 at which they differ. The zoomer bounds $\bar{X}'_n$ will be same for the two sequences at all times $n \leq n_0$, so in particular, the best bounds at time n, denoted by L'_n and U'_n, will be the same up to time n_0. At time n_0, the new bound $\bar{X}'_{n_0}$ will be a lower bound when decoding one of the sequences, and an upper bound when decoding the other. The new bound is either outside or inside the interval (L'_{n_0-1}, U'_{n_0-1}). If it is outside, the new bound cannot be consistent with decoding both sequences. Since the codeword decoding is consistent by definition, the decoding of the noncodeword must be inconsistent. If the new bound is inside the interval limited by the best bounds at time $n_0 - 1$, it will be consistent with these bounds when decoding both sequences, but at time n_0, the decoding results in the interval splitting into two disjoint intervals, $(L'_{0-1}, \bar{X}'_{n_0})$ and $(\bar{X}'_{n_0}, U'_{n_0-1})$. It is thus impossible that the two sequences are decoded to the same interval. In both cases, the assumptions are violated.

Appendix B
Performance Measures

1. Ideal Conditions

The number of possible codewords as X sweeps over the dynamic range D is denoted by C, and the decoder estimate as a function of the random variable X is denoted by $\hat{X}$. The decoder estimate of the input giving rise to the ith codeword is denoted by $\hat{X}_i$; recall that the estimate $\hat{X}_i$ is found as the average of the interval bounds determined by the zoomer. Finally, we denote the interval corresponding to the ith codeword by I_i, and its width by d_i.

We assume that the constant input X is uniformly distributed on the dynamic range D. This is an analytical convenience, but it is not cardinal to our technique. Any piecewise continuous probability density function for X on D can be incorporated in the analysis below. The performance measures are defined as follows:

1) The MSE is defined as

$$\text{MSE} \triangleq E[(X - \hat{X})^2]. \tag{31}$$

To obtain a more operational expression for the MSE, the contribution from the ith interval is given by

$$\text{MSE}_i = E[(X - \hat{X})^2 | X \in I_i] = \sum_{i=1}^{C} \frac{d_i^2}{12}.$$

The total MSE is found by taking the weighted sum of these errors,

$$\text{MSE} = \sum_{i=1}^{C} \frac{d_i}{|D|} \cdot \text{MSE}_i = \sum_{i=1}^{C} \frac{d_i^3}{24Kb} \tag{32}$$

where $|D|$ is the width of the dynamic range. The average input power is

$$E[X^2] = \int_{-Kb}^{+Kb} \frac{1}{2Kb} x^2 \, dx = \frac{(Kb)^2}{3}.$$

Defining SNR $= 10 \log_{10} [E(X^2)/\text{MSE}]$, we thus have

$$\text{SNR} = 10 \log_{10} \frac{8(Kb)^3}{\sum_{i=1}^{C} d_i^3}. \tag{33}$$

2) The worst case error is an important performance measure for nonuniform quantizers such as $\Sigma\Delta$ modulators, since it specifies the local performance, as opposed to the MSE which is a global or average measure. The worst case error is defined as

$$\epsilon \triangleq \max_{X \in D} |X - \hat{X}| = \max_{1 \le i \le C} \left\{ \frac{d_i}{2} \right\} \tag{34}$$

where the last equality follows because the zoomer chooses the quantization interval midpoints as its estimates. The worst case resolution in bits is

$$R = \log_2 \frac{|D|}{2\epsilon} = \log_2 \frac{Kb}{\epsilon}. \tag{35}$$

2. *Nonidealities*

When nonidealities are present, it is necessary to generalize the expressions for MSE and worst-case resolution given in Appendix B-1 which are derived under the assumption that the original zoomer estimates are the midpoints of the corresponding intervals; we first introduce some additional notation. Note that we use the term "actual" to refer to characteristics of a nonideal encoder. We denote the actual input interval corresponding to the ith codeword of a nonideal encoder by I_i, and its width by d_i. The actual lower and upper bounds on the ith interval are denoted by l_i and u_i, respectively. Under ideal circumstances, L and U in the zoomer algorithm correspond exactly to l_i and u_i, and $\hat{X}_i = (l_i + u_i)/2$. Finally we define $q_i = \hat{X}_i - l_i$, $r_i = u_i - \hat{X}_i$ as the signed distances from the original decoder estimate to the edges of the actual interval. If the original zoomer estimate $\hat{X}_i$ is outside of the corresponding actual interval, one of these distances is negative.

The general definitions (31) and (34) of MSE and worst case resolution remain valid, but the MSE contribution from the ith quantization interval, previously given by (32), is modified to

$$\text{MSE}_i = E[(X - \hat{X})^2 | X \in I_i] = \sum_{i=1}^{C} \frac{q_i^3 + r_i^3}{3(q_i + r_i)}.$$

The worst case error, previously given by (34), changes into

$$\epsilon = \max_{1 \le i \le C} \{\max \{q_i, r_i\}\}.$$

These generalizations are reflected in an obvious way in the expressions (33) and (35).

Appendix C
Form of Nonidealities

The discrete-time integrator in Fig. 1 in practice has a transfer function that can be modeled as

$$g \frac{z^{-1}}{1 - \alpha z^{-1}} \tag{36}$$

where g is the gain and α is the leak factor; both these are nominally 1.

The gain is usually determined as a capacitor ratio. The leak factor α in (36) is the result of finite op-amp gain; if the integrator op-amp gain is A, the leak factor is given by

$$\alpha = A/(A + 1) \approx 1 - 1/A.$$

Because we are considering constant inputs or very low-bandwidth applications, large-gain op-amps are probably more easily attainable than in signal acquisition applications.

The zoomer algorithm assumes that the initial integrator states U_0, V_0, etc., are set to zero at the beginning of each conversion cycle. In practice, this can only be achieved with finite precision.

In practice, the 1-b quantizer Q in (1) may not switch between the output values $-b$ and $+b$ at exactly the input value $U = 0$. The 1-b quantizer is more precisely characterized by a DC offset q_0:

$$Q(U) = \begin{cases} -b & \text{if } U \le q_0 \\ +b & \text{if } U > q_0. \end{cases} \tag{37}$$

Real quantizers may also exhibit hysteresis, as in

$$Q_{\text{hyst}}(U_n) = Q[U_n + h_0 Q_{\text{hyst}}(U_{n-1})] \tag{38}$$

where h_0 is a measure of the hysteresis effect. Note that h_0 introduces additional memory in the system. Simulations for the single loop encoder indicate that the performance degradation in the presence of hysteresis is close to the degradation for nonzero initial states. Compensating the zoomer for known hysteresis is simple. Hysteresis is not further treated in this paper.

In practice, circuit and external noise must be taken into account when evaluating the performance of decoders. We consider noisy inputs of the form

$$X_i = X + N_i \tag{39}$$

where X is a constant, and $\{N_i\}$ is a sequence of independent random variables uniformly distributed on the interval $(-Mb, +Mb)$. M is a constant. The optimal procedure in the presence of this type of noise is difficult to determine analytically. However, a simple way to take noise into account is described in Section III-A5.

Appendix D
Nonidealities for Two Stage Modulator

This Appendix describes ways of compensating for known nonidealities. For brevity, curves describing the effects of unknown nonidealities are not included, but may be found in [10] and results are summarized in Table II.

1. Nonunity Integrator Gains

Denoting the integrator gains of the first and second stage by g_1 and g_2, respectively, the difference equations can be written

$$U_n = U_{n-1} + g_1(X_{n-1} - Q(U_{n-1}))$$

$$V_n = V_{n-1} + g_2(-U_{n-1} + Q(U_{n-1} - Q(U_{n-1}))),$$

$$n \geq 1.$$

Solving these with the definitions of Section II-C1 yields the zoomer bounds

$$\overline{X}_n^{(1)} = \frac{1}{n} S_n, \qquad \overline{X}_n^{(2)} = \frac{g_1 W_n + (1 - g_1) S_n - T_n}{\frac{1}{2} n(n-1) g_1}.$$

It is seen that g_2 does not appear in the equations. If the first integrator gain is known, it can be compensated for using (D.1).

2. Leaky Integrators

We make the assumption that the two integrators have the same leak factor α, that is, that the op-amps have the same gain. The difference equations then become

$$U_n = \alpha U_{n-1} + X_{n-1} - Q(U_{n-1})$$

$$V_n = \alpha V_{n-1} + Q(U_{n-1}) - U_{n-1} - Q(V_{n-1}),$$

$$n \geq 1.$$

Defining $S_0 = T_0 = W_0 \triangleq 0$ and the running sums

$$S_n = \sum_{i=0}^{n-1} \alpha^{n-i-1} Q(U_i), \qquad n \geq 1$$

$$T_n = \sum_{i=0}^{n-1} \alpha^{n-i-1} Q(V_i), \qquad n \geq 1$$

$$W_n \sum_{i=0}^{n-1} [n - i - (1 - \alpha)] \alpha^{n-i-2} Q(U_i), \qquad n \geq 1$$

we then have the recursions

$$S_n = \alpha S_{n-1} + Q(U_{n-1}), \qquad n \geq 1$$

$$T_n = \alpha T_{n-1} + Q(V_{n-1}), \qquad n \geq 1$$

$$W_n = \alpha W_{n-1} + S_n + (1 - \alpha) S_{n-1}, \qquad n \geq 1.$$

The zoomer bounds change into

$$\overline{X}_n^{(1)} = \frac{1 - \alpha}{1 - \alpha^n} S_n,$$

$$\overline{X}_n^{(2)} = \frac{(1 - \alpha)^2}{1 + (n-1)\alpha - n\alpha^{n-1}} [W_n - T_n].$$

Similarly to the double loop case, we will consider the effect of leak in the second integrator separately. The difference equations for this situation are

$$U_n = U_{n-1} + X_{n-1} - Q(U_{n-1})$$

$$V_n = \alpha V_{n-1} + Q(U_{n-1}) - U_{n-1} - Q(V_{n-1}),$$

$$n \geq 1.$$

Defining $S_0 = T_0 = W_0 \triangleq 0$ and the running sums

$$S_n = \sum_{i=0}^{n-1} Q(U_i), \qquad n \geq 1$$

$$T_n = \sum_{i=0}^{n-1} \alpha^{n-i-1} Q(V_i), \qquad n \geq 1$$

$$W_n = \sum_{i=0}^{n-1} \frac{1 - \alpha^{n-i}}{1 - \alpha} Q(U_i), \qquad n \geq 1$$

we have the recursions

$$S_n = S_{n-1} + Q(U_{n-1}), \qquad n \geq 1$$

$$T_n = \alpha T_{n-1} + Q(V_{n-1}), \qquad n \geq 1$$

$$W_n = \alpha W_{n-1} + S_n, \qquad n \geq 1.$$

The zoomer bounds change into

$$\overline{X}_n^{(1)} = \frac{1}{n} S_n,$$

$$\overline{X}_n^{(2)} = \frac{(1 - \alpha)^2}{\alpha^n - n\alpha + (n-1)} [W_n - T_n].$$

From the above discussion we see that if the integrator leaks are known, they can be compensated for. Simulations indicate that at an OSR of 127, the sensitivity of the original zoomer algorithm towards the second integrator gain is on the order of 50 times smaller than that towards the first integrator leak [10].

3. Nonzero Initial States

For non-zero initial integrator states U_0, V_0, the zoomer bounds become

$$\overline{X}_n^{(1)} = \frac{-U_0 + S_n}{n},$$

$$\overline{X}_n^{(2)} = \frac{-nU_0 + V_0 + W_n - t_n}{\frac{1}{2} n(n-1)} \tag{40}$$

with S_n, T_n, W_n defined by (18)–(20). Note that in (40), U_0 is more important than V_0, again confirming that nonidealities closer to the input summing node have greater impact on performance. If the initial states are known, they can be compensated for using (40).

4. DC Offset in Quantizers

Assuming that both quantizers are characterized by the same DC offset q_0, the zoomer bounds become

$$\overline{X}_n^{(1)} = \frac{q_0 + S_n}{n},$$

$$\overline{X}_n^{(2)} = \frac{-q_0 + W_n - T_n}{\frac{1}{2} n(n-1)}. \tag{41}$$

Note the similarities and differences between the above equations and (40) for nonzero initial states. The analogies with nonzero initial states indicate that an offset in the first quantizer is comparatively more detrimental to performance. If the DC offsets are known, they can be compensated for using (41).

5. Random Noise

Following the derivations of previous sections, a simple countermeasure against white uniform noise on the interval $(-Mb, +Mb)$ is to change the zoomer bounds to

$$\overline{X}_n^{(1)} = \frac{1}{n} S_n - 2\frac{MQ(U_n)}{\sqrt{3n}} \tag{42}$$

$$\overline{X}_n^{(2)} = \frac{W_n - T_n}{\frac{1}{2}n(n-1)} + 2\sqrt{\frac{2(2n-1)}{n(n-1)}}\frac{MQ(V_n)}{3}. \tag{43}$$

In (42), the sign of the noise correction is chosen opposite that of $Q(U_n)$, whereas in (43), it is chosen to be the same as that of $Q(V_n)$. In both cases, this leads to weaker bounds.

References

[1] R. M. Gray, "Oversampled sigma-delta modulation," *IEEE Trans. Commun.*, vol. COM-35, no. 5, pp. 481–488, May 1987.

[2] J. C. Candy, Y. C. Ching, and D. S. Alexander, "Using triangularly weighted interpolation to get 13-bit PCM from a sigma-delta modulator," *IEEE Trans. Commun.*, vol. COM-24, no. 11, pp. 1268–1275, Nov. 1976.

[3] J. C. Candy, "A use of limit cycle oscillations to obtain robust analog-to-digital converters," *IEEE Trans. Commun.*, vol. COM-22, no. 3, pp. 298–305, Mar. 1974.

[4] J. C. Candy, "A use of double integration in sigma delta modulation," *IEEE Trans. Commun.*, vol. COM-33, no. 3, pp. 249–258, Mar. 1985.

[5] J. C. Candy, "Decimation for sigma delta modulation," *IEEE Trans. Commun.*, vol. COM-34, no. 1, pp. 72–76, Jan. 1986.

[6] S. Hein and A. Zakhor, "Lower bounds on the MSE of the single and double loop sigma delta modulators," in *Proc. Int. Conf. Circuits Syst.*, May 1990, pp. 1751–1755.

[7] S. Hein, K. Ibraham, and A. Zakhor, "New properties of sigma delta modulators with DC inputs," *IEEE Trans. Commun.*, vol. 40, no. 8, pp. 1375–1387, Aug. 1992.

[8] A. Zakhor and S. Hein, "Optimal decoding for sigma delta modulators," U.S. Patent 5164727, Nov. 17, 1992.

[9] K. C.-H. Chao, S. Nadeem, W. L. Lee, and C. G. Sodini, "A higher order topology for interpolative modulators for oversampling A/D converters," *IEEE Trans. Circuits Syst.*, vol. 37, no. 3, pp. 309–318, Mar. 1990.

[10] S. Hein and A. Zakhor, "Optimal decoding for data acquisition applications of sigma delta modulators," ERL Memo, Univ. California, Berkeley, Aug. 1991.

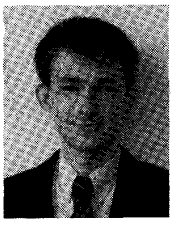
[11] R. M. Gray, "Spectral analysis of quantization noise in a single-loop sigma-delta modulator with dc input," *IEEE Trans. Commun.*, vol. 37, no. 6, pp. 588–599, June 1989.

[12] P. W. Wong and R. M. Gray, "Two-stage sigma-delta modulation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 11, pp. 1937–1952, Nov. 1990.

[13] K. Uchimura, T. Hayashi, T. Kimura, and A. Iwata, "Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 12, pp. 1899–1905, Dec. 1988.

[14] W. Chou, P. W. Wong, and R. M. Gray, "Multistage sigma-delta modulation," *IEEE Trans. Inform. Theory*, vol. 35, no. 4, pp. 784–796, July 1989.

[15] P. F. Ferguson, Jr., A. Ganesan, and R. W. Adams, "One bit higher order sigma-delta converters," in *Proc. Int. Symp. Circuits Syst.*, May 1990, pp. 890–893.



Søren Hein (S'88–M'93) was born in May 1968 in Copenhagen, Denmark. He received the M.Sc. degree in electrical engineering from the Technical University of Denmark in January 1989 and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in June 1992.

He currently works for Siemens AG in Germany. His research interests include algorithmic aspects of oversampled A/D conversion and signal processing. He has also worked on error-correction coding for satellite communications.

Avideh Zakhor (S'87–M'87) received the B.S. degree from the California Institute of Technology, Pasadena, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, all in electrical engineering, in 1983, 1985, and 1987, respectively.

In 1988, she joined the Faculty at the University of California, Berkeley, where she is currently Assistant Professor in the Department of Electrical Engineering and Computer Sciences. Her research interests are in the general area of signal processing and its applications to images and video, and biomedical data. She has been a consultant to a number of industrial organizations in the areas of signal processing, communications, and medical imaging, and has two pending patents on MRI signal processing, one on sigma delta modulators, and one on phase shifting mask design for optical lithography.

Dr. Zakhor was a General Motors scholar from 1982 to 1983, received the Henry Ford Engineering Award and Caltech Prize in 1983, was a Hertz Fellow from 1984 to 1988, received the Presidential Young Investigators (PYI) Award, IBM Junior Faculty Development Award, and Analog Devices Junior Faculty Development Award in 1990, and ONR Young Investigator Award in 1991. She is a member of Tau Beta Pi, Sigma Xi, and the IEEE SP Society.