CR115460

Delta Modulation

Final Report

January 1, 1971 - December 31, 1971

National Aeronautics and Space Administration

MSC - Houston

under

NASA GRANT NGR 33-013-063



DEPARTMENT OF ELECTRICAL ENGINEERING



# THE CITY COLLEGE RESEARCH FOUNDATION

THE CITY COLLEGE of
THE CITY UNIVERSITY of NEW YORK

Donald L. Schilling
Professor
Principal Investigator

# Delta Modulation

Final Report

January 1, 1971 - December 31, 1971

National Aeronautics and Space Administration

MSC - Houston

under

NASA GRANT NGR 33-013-063

DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling

Donald L. Schilling

Professor

Principal Investigator

Table of Contents

Introduction

This final report summarizes the research sponsored by the National Aeronautics and Space Administration under NASA Grant NGR 33-013-063 for the period January 1, 1971 through December 31, 1971. The research supported by this grant encompasses the problems of source encoding using delta modulation.

Part I of this report presents the conclusions of the research conducted into the design of the Song Adaptive Delta Modulator for source encoding voice signals. Here we show the variation of output SNR vs input signal power when 8, 9 and 10 bit internal arithmetic is employed. We are currently using NASA supplied voice intelligibility tapes to test the 10-bit system. When the recordings are complete the resulting tapes will be sent to NASA for analysis. In addition, we have supplied NASA with a detailed schematic of the 10-bit adaptive DM and they are proceeding to construct the device.

Part II of this report presents an analysis of a delta modulator designed to minimize the in-band rms error. This is accomplished by frequency shaping the error signal in the modulator prior to hard limiting. The result of this, is a significant increase in the output SNR measured after low pass filtering.

One of the advantages of this delta modulator is that the receiver is a simple integrator (low pass filter). This should be compared to the Song delta modulator where the adaptive feature is built into the feedback path.

New Areas of Research are discussed in Part III. In particular we are shifting our attention to the use of delta modulation to source encode video signals. Several different types of delta modulators will be considered which will combine the wide dynamic range of the Song-type DM with the greatly increased output SNR possible using the Nth-order DM and the In-Band Minimum rms DM.

The results presented in this report represent a significant step forward in the design of delta modulators. The research supported by

this grant has resulted in two PhD Dissertations, listed in Part IV, and in the publication of several papers (Part V).

Participating in this program were:

Drs. A. S. Rosenbaum, C. L. Song, and J. Garodnick

and

Messrs, S. Altman, J. Angermaier, T. Cassa, and R. Soicher

I. An Adaptive Robust Delta Modulator

The Adaptive Robust Delta Modulator conceived by Song and implemented digitally by Garodnick has been described in detail in the semi-annual report presented to NASA in June 1971. A paper covering this research has been published in the special issue of the IEEE Transactions on Communication Technology entitled "Digital Processing of Signals", December 1971. In addition, a detailed schematic of the 10-bit system has been given to B. Batson of MSC - Houston.

Dynamic Range Improvement Using an Increased Number of Bits

The delta modulator was constructed digitally so that the complete system could be integrated. In addition, digital implementation of a system is a more precise procedure than analog implementation since the system characteristics are known precisely.

A major limitation of a digital system is that a voltage must be quantized to represent it digitally. Thus it can have $2^N$ possible values if N bits are employed. Further, addition, multiplication and division accuracy is limited by the finite number of bits used in the internal system arithmetic. The number of bits used is usually greater than or equal to N. In our system we used N-bits internally as well as for A/D and D/A conversion.

The system was constructed using 8, 9, and 10 bit arithmetic. Plots of output SNR vs input signal power were made for the Song system ($\alpha=1$, $\beta=\frac{1}{2}$) and for the Enhanced Abate system ($\alpha=1$, $\beta=0$). These results are shown in Figs 1 and 2.

Referring to these figures we see that the use of an increased number of bits significantly widens the dynamic range although the maximum output SNR is somewhat reduced. The results were taken for an input sinusoid of 400 Hz, sampled at 56 kHz. Other ratios $f_s/f_M$ could be employed, however, the tests were performed to check dynamic range not maximum SNR. In this regard we note that the

Enhanced Abate scheme ($\alpha$=1, $\beta$=0) suffers a 2dB loss in maximum SNR but an increase in dynamic range of 11dB, while the Song scheme ($\alpha$=1, $\beta=\frac{1}{2}$) suffers a 2.5dB loss in maximum SNR but an increase in dynamic range of 14dB. The Song dynamic range is 40dB, when measured at the 3dB points.

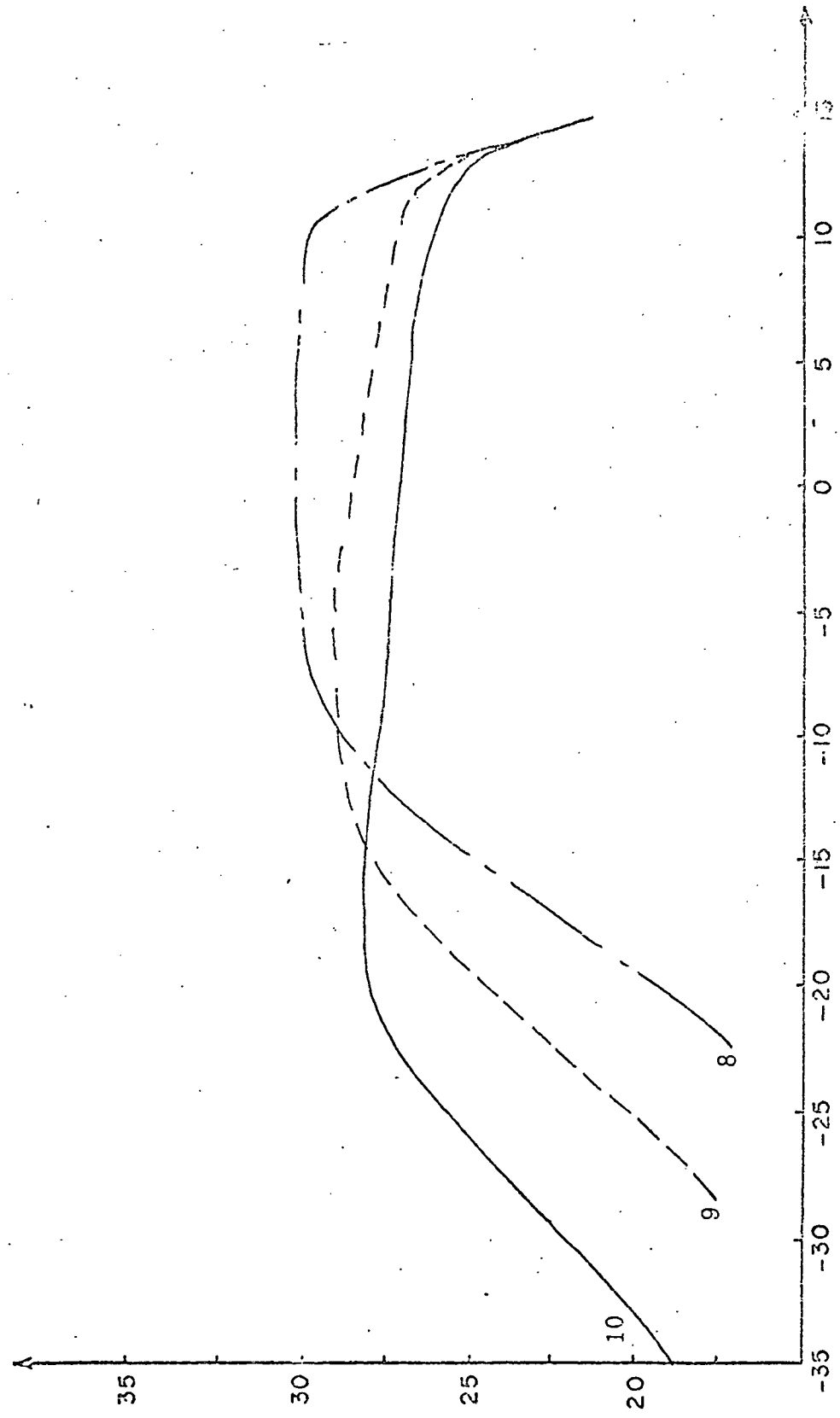



Figure 1 The Song Delta Modulator ($\alpha = 1$, $\beta = \frac{1}{2}$)

$f_M$ = 800 Hz

$f_S$ = 56 kHz

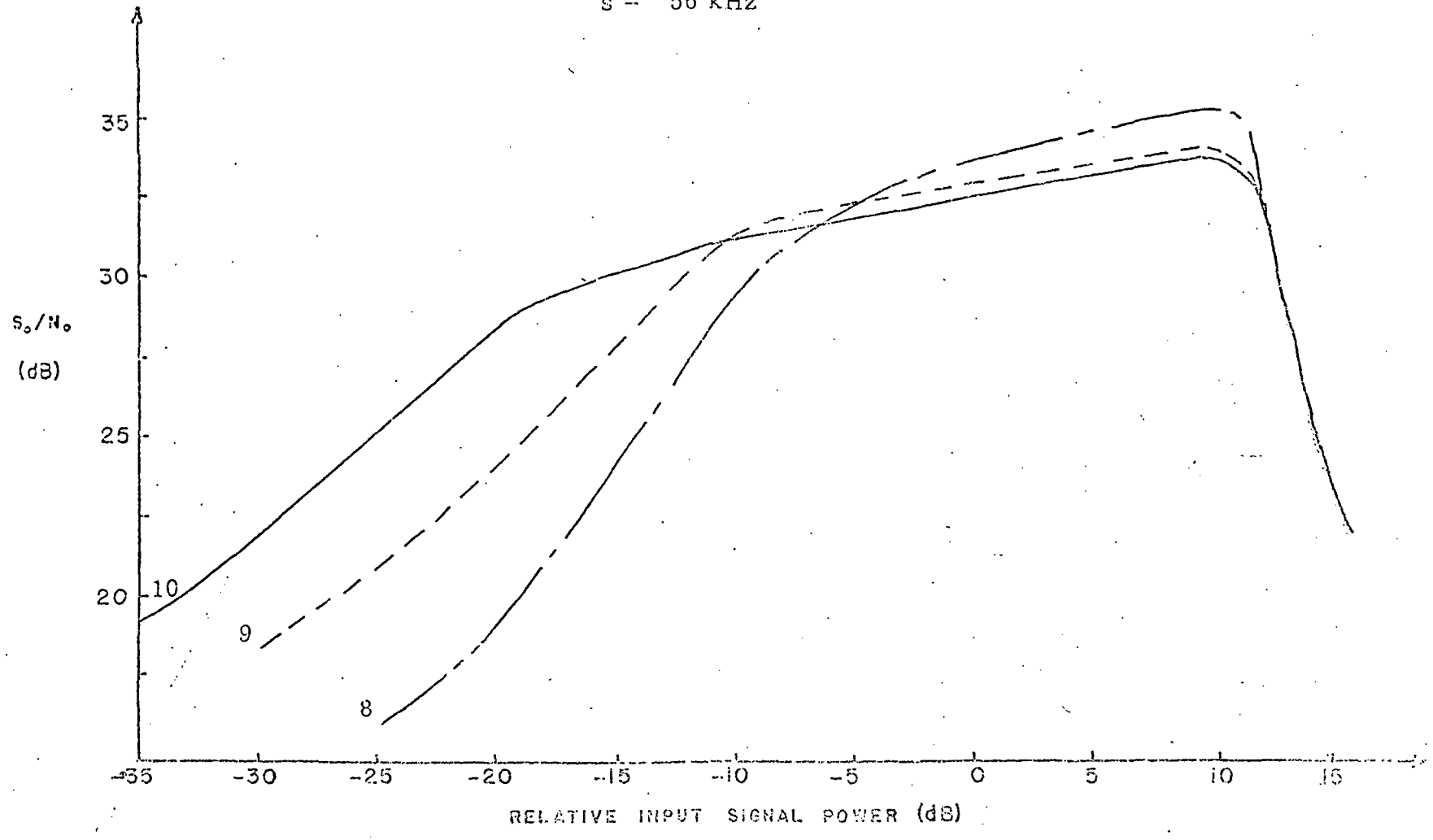



Figure 2 The Enhanced Abate Delta Modulator ($\alpha = 1$, $\beta = 0$)

## II. The Least In-Band Mean Square Error Delta Modulator

### Summary

The Least In-Band Mean Square Error DM is a source encoder which shapes the error signal just prior to hard-limiting. This shaping is illustrated in Fig 1. Here we see that the simple "linear" delta modulator is modified using a "shaping" filter. Note that the feedback network is not affected and hence the receiver is the integrator (low pass filter) used in the linear delta modulator.

Figure 2 shows spectral density curves for a typical sentence of speech. The ordinate is the spectral density in dB while the abscissa is frequency in kHz. Figure 2A is the spectrum of a linear DM. Note that spectrum decreases slighly with increasing frequency and that over a 5kHz band, the average of the peaks is approximately -43dB. Figure 2B is the spectrum of the "shaped-error" DM using a block -1 configuration (see Chapter 5 for a description of the system). Note that the spectral density is greatly suppressed in the 0-5kHz band. The average of the peaks now being approximately -55dB. This results in a SNR improvement over the 5 kHz band of 12dB.

The following discussion is a PhD Dissertation submitted by A. S. Rosenbaum to the Faculty of Engineering at the City College of New York.

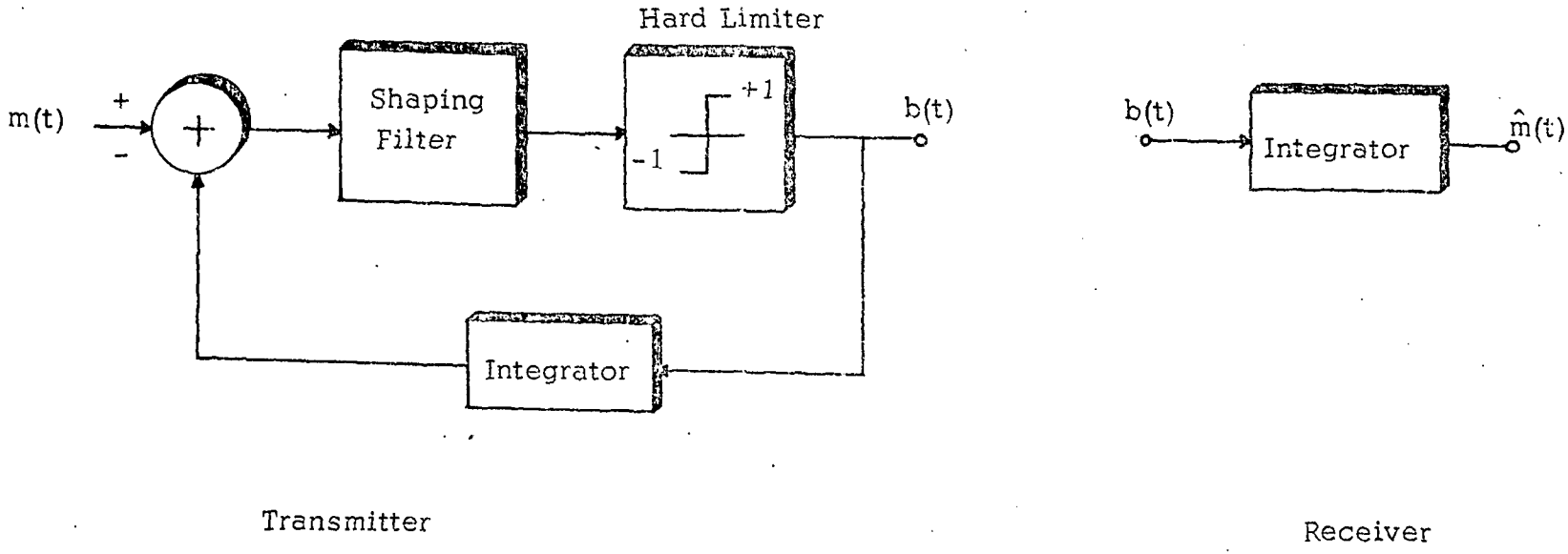



Fig 1 The Least In-Band Mean Square Error Delta Modulator

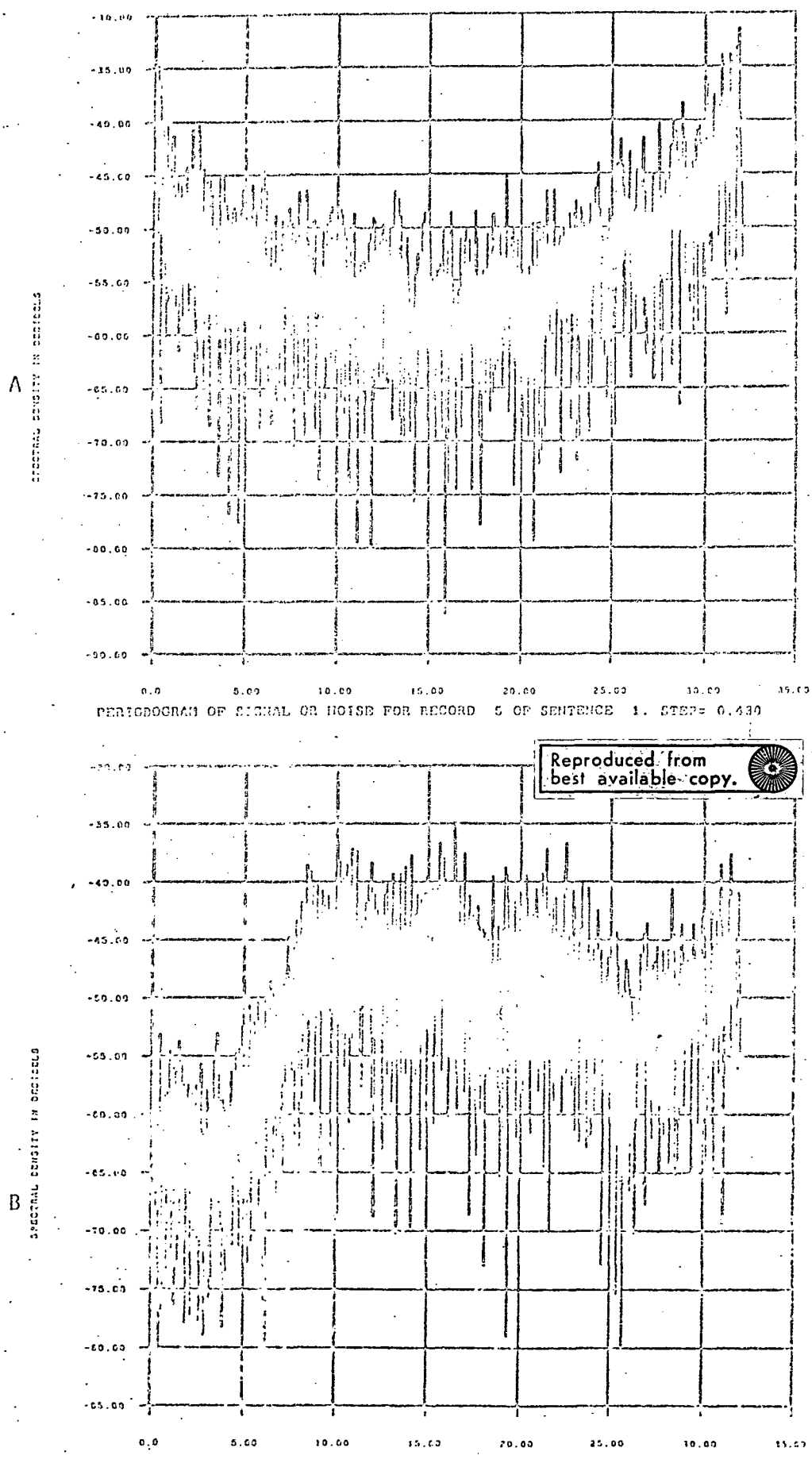



FIG. 2. NOISE SPECTRUM, RECORD 5, STEP=.43
A. STANDARD DM B. BLOCK-1, M=8

# Chapter I

## 1.1 Introduction

This dissertation deals with the problem of representing an analog signal, ie, a time continuous low pass function such as speech, by a digital sequence, and then reconstructing from that sequence an analog waveform which is a suitable approximation to the original. This so-called analog source encoding problem has become very important since the tremendous growth of digital processing and transmission of information, which was made possible by significant advances in hardware technology. Since now the very complex devices required to implement a reasonably efficient source encoding-decoding scheme are rapidly becoming feasible, research toward improved techniques is of great practical as well as academic interest.

In general, when an analog message is reduced to a form which is discrete in both time and amplitude, some information is lost and the departure of the reconstruction from the original, called quantization noise, is an unavoidable result. The amount and nature of the quantization noise, however, depends greatly on the rule for choosing the digital sequence and the rule whereby the analog approximation is formed from that sequence, ie, the encoding and decoding.

Quite a lot of research, albeit in a fairly short time span, has already been done in this area. Much of it, understandably, has concentrated on one or another sub-problem with tractable sets of constraints. For example, a Gaussian Markov source and a minimum mean squared error (MMSE) distortion measure will allow the application of known theory, and nearly optimum encoders have been suggested for this case. On the other hand, for the much larger class of sources encountered in practice, and especially distortion measures more meaningful than MSE, very little has been done in comparison with the size of the problem.

Aside from the obvious practical considerations, there remains at least one other factor which motivates continued research in an already heavily trampled field. It is the absence, except for certain special cases, of a general theory which leads the way to the optimum solution, ie, the best encoder-decoder pair for any given source class and fidelity criterion. The limits of achievable performance, however, are known. The source encoding theory which stems from the work of C. E. Shannon and others, sets forth the fundamental limitiations on performance for certain source classes and fidelity criteria when the digital processing is done at a given fixed rate. This rate-distortion theory remarkably provides a knowledge of "Nature's limits"; nevertheless, for all its transcendental reasoning and elegant mathematics, it does not tell how to achieve these

limits in practice. Because the attendant theory is largely not constructive in that sense, source encoding for the most part is still as much art as science.

Most encoding techniques, especially the adaptive algorithms, are of the ad-hoc variety, suggested by a combination of engineering experience and intuition rather than a canonic realization known a-priori to be optimal. The point to be made, though, is that the lack of restrictions on the form of a solution offers the freedom to explore in many directions. Unlike research in some other areas where, starting with axioms, theorems may be proved and results obtained thereby, the present problem requires a more empirical approach.

All the aforementioned reasons have combined to stimulate this investigation. Its major result is a method of encoding that focuses attention on the spectral properties of the quantization noise, thereby encouraging the application of a much more meaningful fidelity measure: frequency weighted noise power. This indicator reflects the impairment caused by quantization noise via an average

$$d_{WN} = \int \Omega_q(\omega) W(\omega) d\omega \qquad (1.1.1)$$

where $\Omega_q$ is the spectral density function of the noise process q, and $W(\omega)$ is a nonnegative weighting function describing the relative degradation contributed by noise concentrated in a narrow band at frequency $\omega$.

This new method utilizes delay and memory at the encoder to make digital decisions in blocks, with the intent of controlling not just the mean squared value of the noise, but its entire autocorrelation structure. Nearly all prior methods adopt the MMSE philosophy. This is a good first step, and much more amenable to analysis. However, it will be shown that so-called weighted square error criteria (of which MSE is a trivial special case) not only leads to better performance, but lends valuable insight into the design of encoder algorithms.

This theory does not give the decoder design, unfortunately. A decoder algorithm must be assumed. Once this is done, however, that encoder decision strategy is easily found which causes the reconstruction to follow the source in the best frequency weighted error sense. As discussed later, the improvement afforded thereby is greatest when the sampling rate is high. Therefore the bulk of this work is concerned with applying the encoding technique to delta modulation (DM), which has the highest sampling rate of any common system. Nevertheless, the theory is quite general and easily applied to other types of encoders as well.

The remainder of this chapter provides the necessary background and nomenclature to follow the theoretical discussion of the encoding procedure given in Chapters 2-4. Computer simulation was used to test the performance of the encoder algorithm as applied to DM, with

various block sizes. Digitally recorded speech served as the main source material, and with the aid of special equipment the reconstruction was converted to analog form for audition. The experimental results are reported in Chapter 5. Chapter 6 is devoted to documenting the simulation programs as well as related theory, and to describe the interface equipment used to obtain audio and graphical output.

The quantitative measure of performance is extracted from spectral analysis of the quantization noise sequence. Both sine wave and speech inputs were used. An interesting byproduct of these results is a good picture of the frequency content of the quantization noise of standard DM as the step size is varied from the slope overload region to the granular noise region.

Just as sample by sample encoding is an iterative partitioning of the real line, so does higher dimensional (block) encoding induce a partition of a real N-space into encoding volumes. By comparing, for example, the 2 dimensional decision regions of the weighted noise scheme with the analogous decision volumes obtained by iterating a standard encoder twice, much intution is gained about the way it works and the advantages of a higher dimensional algorithm. In particular, one can see how the future samples interact and how the decision regions are so shaped as to yield an error sequence whose autocorrelation has the sought for properties.

## 1.2 Background

The purpose of this section is to give some feeling for the source encoding problem, and to place the present work in perspective by drawing comparisons between it and pertinent prior contributions. However, except for those references, the usual historical development is omitted in favor of a brief discussion of the inherent philosophies of the current major encoding methods. It is felt that more would be gained by considering underlying principles instead of enumerating the specific features of various popular systems, many of which are more or less variations on the same theme.

Digital waveform transmission traces its roots to the sampling theorem, which says that a bandlimited function is completely specified by an infinite sequence of its samples. In practice, analog messages are not strictly bandlimited (for one thing, they have finite duration), and bandlimited reconstruction would require infinite delay. Nevertheless, the degradation caused by finite delay reconstruction of essentially bandlimited functions is usually negligible as compared with the impairment brought about by digital encoding. Therefore, the conceptually narrower problem of encoding and reproducing real valued sequences will be considered equivalent to that of digitally encoding and reproducing a continuous time source.

A sequence will be denoted by the member character underscored, where the presence of a subscript or superscript

signifies truncation forward or backward in index, respectively. For example, the sequence of operation instants from the infinite past up to the most recent, $\{\ldots t_{n-1}, t_n\}$ is denoted $\underline{t}_n$, and the doubly truncated sequence of source samples $\{s_i, \ldots, s_n\}$ is written compactly as $\underline{s}_n^i$.

An incisive approach to sequence encoding was taken by Fine[1], whose elegant formulation of the problem is paraphrased below in order to facilitate the ensuing discussion of encoding methods. A digital system, as depicted in Figure 1, comprises three components: the encoder, a channel for transmission or storage of the digital sequence, and the decoder. Each component is viewed as a causal transformation which maps, at discrete instants of operation, the sequence at its input into an output value.

The encoder, T, maps the real value source samples $\underline{s}_n$ into a K-ary valued symbol $c_n$, i.e., an element of the alphabet $\{\alpha_1, \ldots, \alpha_K\}$. This is expressed in functional form as

$$c_n = T(\underline{s}_n) \qquad (1.2.1)$$

The possibility of digital errors, perhaps resulting from transmission over a real channel, is accounted for by the channel transformation, C, being a random mapping which is governed by a prescribed set of transition probabilities.

For the present, an ideal channel will be assumed, so let C be the identity mapping. The decoder is a transformation R relating the channel sequence to the real valued reconstruction $\hat{\underline{s}}$ which is written as

$$\hat{s}_n = R(\underline{c}_n) \tag{1.2.2}$$

Thus at the $n^{th}$ instant of operation 1.2.1 and 1.2.2 take place in that order, but in effect simultaneously. Any delays inherent in a physical realization of the process are unimportant here and will be ignored. Note that T and R may depend on the entire past of their respective inputs, i.e., complete memory, which allows for the utilization of all the information available to the system at each moment.

The distortion measure adopted by Fine is the so-called class of mean $\phi$ criteria

$$d_\phi = E\{\phi(s_{n+\varepsilon} - \hat{s}_n)\} \tag{1.2.3}$$

where the expectation is an ensemble average over all possible input and output sequence pairs. The cost function $\phi$ is positive, even, and convex, examples of which are the commonly used absolute value and square. The value of e determines whether the problem is one of interpolation, $\varepsilon < 0$, prediction $\varepsilon > 0$, or synchronous reconstruction $\varepsilon = 0$.

Concentrating on the cases $\varepsilon \leq 0$ with the quadratic cost function, Fine derived necessary conditions for a joint optimization of T and R. However, this leads to complicated algorithms wherein the mappings may be functions of time, and the statistical parameters of the

source must be known. Continuing along the same lines, Gish[2] considered the more practical suboptimum solution obtained when the mappings are required to be time invariant.

While the present work differs from [1] and [2] in several important respects, it has been influenced by these two prior contributions in the areas of problem definition and the philosophy of encoding; in particular, the concept that a function of the error variable directly partitions the source space into locally optimum encoding regions.

Turning now to some basic considerations of digital encoding design, assume (with no loss of generality) that the operation instants are uniformly spaced in time,

$$t_n - t_{n-1} = \tau, \quad n = 0, \pm 1, \pm 2... \qquad (1.2.4)$$

so that the system processes at the rate*

$$\rho = \tau^{-1} \log_2 K \qquad (1.2.5)$$

bits/second. Keeping $\rho$ fixed, which is a major constraint of the problem, establishes a trade-off between the rapidity of processing samples (and generating reconstruction values) and the amount of information which may be conveyed at each instant of operation, or with each $c_n$.

---

* This is not to be confused with the information theoretic rate.

The relative magnitudes of $\tau^{-1}$ and K are a key factor in categorizing, and comparing, the various common encoding methods.

The simplest example, memoryless amplitude quantization, as used in pulse code modulation (PCM) systems, utilizes the minimum possible sampling rate and a large alphabet to reconstruct each sample independently with a high degree of precision. Much work has previously been done on the problem of obtaining mappings such that

$$E\{(s-\hat{s})^2\} = E\left\{\left[s - R(T(s))\right]^2\right\} \qquad (1.2.6)$$

is minimized.

At the other extreme in the range of sampling rate is DM, which operates at a substantial multiple of (typically 4 to 10 times) the Nyquist rate, but uses only a binary encoding. Neighboring samples, being so closely spaced in time, cannot differ greatly as a consequence of the band limited property of the source. It is advantageous, then, to encode only correctional or difference information in order to update the reconstruction from the present value to the next. Since this difference is usually small relative to the r.m.s. level of the source, giving the difference a binary, or two state, representation is an acceptable method of encoding.

When couched in the framework of random process theory, DM is an unsophisticated member of the large class

of predictive encoding schemes which exploit redundancy in the stochastic source sequence. Heuristically speaking, at the n-th cycle $s_n$ is trivially predicted by the decoder to equal $s_{n-1}$. The error in that prediction, when revealed at the encoder by examining $s_n$, is binary encoded and $\hat{s}_n$ is so obtained as a corrected prediction:

$$\hat{s}_n = s_{n-1} + R\left(T(s_n - \hat{s}_{n-1})\right) \tag{1.2.7}$$

Depending on the autocorrelation of the source, a significant refinement may be made if more memory is employed, as for example the linear prediction (of which DM is a special case)

$$\hat{s}_n(\text{predicted}) = \sum_{j=1}^{J} h_j \hat{s}_{n-j} \tag{1.2.8}$$

which gives the decoder and encoder relationships

$$\hat{s}_n = \sum_{j=1}^{J} h_j \hat{s}_{n-j} + R(c_n) \tag{1.2.9}$$

$$c_n = T\left(s_n - \sum_{j=1}^{J} h_j \hat{s}_{n-j}\right) \tag{1.2.10}$$

Better prediction has the effect of more completely extracting redundancy from the source to minimize the amplitude of the prediction error, which is identically the quantization noise. On the other hand, its effectiveness is tied to a good alignment of the predictor to the statistical (second order) properties of the source, ie, spectral density in the

stationary case. Many sources of practical interest, such as speech, are decidely nonstationary, so that to be effective the system must also be adaptive in the sense that it dynamically tracks the source parameters. Such a system clearly requires a very complex device.

Operating at a sampling rate between the extreme of PCM and DM is the hybrid system called differential PCM, or DPCM. It features both multilevel quantization and prediction. As an example, by doubling $\tau$, $\log_2 M$ may be doubled and the transmission rate in bits/sec is unchanged. For certain sources, it may be desirable to trade the redundancy lost by increasing $\tau$ for the much improved size $K^2$, of the alphabet with which to encode the larger average prediction error. While DPCM affords the opportunity of using just the right mix of prediction and alphabet size to best encode a given source, it gives up the chief advantage of DM, namely the utter simplicity of the encoder and decoder.

At this point it is convenient to establish some basic background in the frequency domain properties of discrete time series, or sequences.[3],[4] As a real valued function on a discrete index set $\{t = 0, \pm\tau, \pm 2\tau, \ldots\}$, a stochastic sequence has no spectrum in any physical sense.

However, an important attribute of one which is wide sense stationary, say $\underline{x}$, is its spectral density function* given by

$$\Omega_x(\omega) = \sum_{k=-\infty}^{\infty} R_{xx}(k\tau)e^{i\omega k\tau} \quad , \quad |\omega| \leq \frac{\pi}{\tau} \qquad (1.2.11)$$

where

$$R_{xx}(k\tau) = E\{x_n x_{n+k}\} = E\{x(n\tau)x(n\tau+k\tau)\} \qquad (1.2.12)$$

is the shift invariant autocorrelation function.

The spectral density function is defined only in the main interval $|\omega| \leq \frac{\pi}{\tau}$, but being a Fourier series the expression (1.2.11) repeats with period $2\pi/\tau$.

The link between this mathematical device and a physical quantity is made by considering the sequence to be the uniformly spaced samples of a continuous time random process, say y(t), which is also wide sense stationary. If the power spectral density of y,

$$\Omega_y(\omega) = \int_{-\infty}^{\infty} R_{yy}(\xi)e^{i\omega\xi}d\xi \qquad (1.2.13)$$

where

$$R_{yy}(\xi) = E\{y(t)y(t+\xi)\} \qquad (1.2.14)$$

vanishes for $|\omega| > \frac{\pi}{\tau}$, then $\Omega_y$ coincides with $\Omega_x$ exactly. That is, the spectral densities of the sequence of samples

* This assumes an absolutely summable autocorrelation sequence if singularities are not allowed.

with interval $\tau$, and that of the sampled random process which is bandlimited to $\frac{\pi}{\tau}$, are equal.

Furthermore, the quasi-physical process

$$z(t) = \sum_{n=-\infty}^{\infty} y(t)\delta(t-n\tau) = \sum_{n=-\infty}^{\infty} x(n\tau)\delta(t-n\tau) \quad (1.2.15)$$

which is an infinite Dirac comb modulated by y has the power spectrum density

$$\Omega_z(\omega) = \sum_{n=-\infty}^{\infty} R_{yy}(n\tau)e^{i\omega n\tau} \quad (1.2.16)$$

so that the impulse train has the spectrum of x repeated.

Note from the above that the reconstruction sequence $\hat{\underline{s}}$ and the quantization noise sequence $\underline{s} - \hat{\underline{s}}$ will have spectra which essentially occupy the radian band $\left(-\frac{\pi}{\tau}, \frac{\pi}{\tau}\right]$, which is larger than the bandwidth of the source by the same amount as the sampling rate exceeds the Nyquist rate.

Now by intergrating the spectral density function* one gets, after the allowed interchange in order,

$$\int_{-\pi/\tau}^{\pi/\tau} f_x(\omega)d\omega = \sum_{-\infty}^{\infty} \int_{-\pi/\tau}^{\pi/\tau} R_{xx}(k\tau)e^{i\omega k\tau}d\omega = \frac{2\pi}{\tau} R_{xx}(0) \quad (1.2.17)$$

Therefore, the mean squared value of a sequence is proportional to the area of its spectral density in the band $|\omega| \leq \frac{\pi}{\tau}$.

---

* This is the analog of Parseval's relation for Fourier series.

The point to be made is that MSE is not necessarily representative of the performance of a system which samples (and reconstructs) faster than the Nyquist rate, since it includes noise power falling outside the bandwidth of the source which will utimately be eliminated by the decoder output filter. When high rate sampling is done, e.g., DM, only a small fraction of the reconstruction bandwidth is occupied by the signal. Thus the shape of the noise spectral density could drastically affect that portion of the MSE which comes from noise actually in the source bandwidth.

This fact was recognized by Cutler[5], who suggested a means whereby the noise spectral density could be shaped advantageously by a method which has become known as noise feedback encoding. Quite simply, if the error sequence is to have its power concentrated at the higher frequencies, it must oscillate more rapidly, that is with a smaller average period. This can be accomplished, as in the Cutler invention, by remembering the immediately previous error, and using that information to bias the present decision so that the error arising from it will tend to be of opposite polarity. Consider, then, merely adding the previous quantizing error to the present sample prior to encoder mapping. If, say, that error were positive,

$$q_{n-1} = s_{n-1} - \hat{s}_{n-1} \geq 0 \qquad (1.2.18)$$

then certainly

$$\hat{s}_n = R\left(T(s_n+s_{n-1}-\hat{s}_{n-1})\right) \geq R\left(T(s_n)\right) \qquad (1.2.19)$$

which causes the n-th quantization error $s_n-\hat{s}_n$ to now be smaller algebraically - more negative. Similarly, a negative error biases the following one toward being positive. This was accomplished by feeding back the previous error to the input - hence the name noise feedback.

This technique was subsequently pursued by Spang and Schultheiss[6] in an attempt to improve PCM. By sampling slightly faster than the Nyquist rate, they argued, noise feedback could be used to force most of the quantization noise into the narrow frequency corridor between the highest signal frequency and $\pi/\tau$.

Their investigation served to place the idea on a more firm analytical foundation, as well as to generalize the correction signal to a linear combination of many past errors. Thus their encoder employed a transversal filter in the feedback path, as given by the encoder relation

$$c_n = T\left[s_n + \sum_{j=1}^{J} H_j[s_{n-j}-\hat{s}_{n-j}]\right] \qquad (1.2.20)$$

Analysis was carried out by making the approximation that if the quantizing was fine enough and overload was rare, then the quantizer could be replaced by a box that merely adds internally an uncorrelated noise sequence to its input, and

the additional assumption that this approximation holds true even after the addition of the feedback loop carrying this noise back around to the quantizer input. This virtually linearized the system so that straightforward analysis techniques could be applied to determine the optimum set of feedback weights $H_j$ which minimizes the in-band noise. The difficulty which accounted for a great deal of their effort was to carry it out subject to overload and other constraints which if violated would nullify the linearizing assumptions.

Based on analytical results, they claimed a 95%, or 13 dB, decrease of in-band noise power with a 25% increase over Nyquist rate in sampling, and a memory length J of 30. This assumes, however, an unbounded number of quantizer levels to assure no overload, although the interlevel spacing is kept the same with and without the noise feedback. Unfortunately, no experimental work was reported.

A concurrent investigation of PCM with noise feedback was conducted at Bell Laboratories by Kimme and Kuo[7], Brainard[8] and others, with particular emphasis on encoding video signals. The system configuration was generalized in their work by the inclusion of a linear predistortion filter at the input and de-emphasis reconstruction filter at the output, in addition to the tapped delay structure in the feedback path, thus making a total of three components to be designed.

In [7] the roles of the pre-emphasis and de-emphasis filters were stressed, and the noise feedback aspect seems to have been downgraded. Perhaps from computational considerations in the design procedures, at most a two section feedback filter was used in the example given, and results were not very dramatic. Even so, improvement of 12 dB in weighted noise over conventional PCM was reported in [8], this coming from the particular inband weighting since there was no oversampling.

The present work indicates that merely two taps in the feedback filter would be ineffective more or less, depending on the actual shape of the weighting function.

Indeed the number of taps needed, as a general rule, varies inversely as the ratio of sampling rate to the Nyquist rate, so that PCM, which has a low ratio, requires the longest filter. It is conjectured that a large portion of that 12 dB improvement derives from the pre-emphasis, rather than the noise feedback. In any event, recent research has shown that efficient encoding of video signals must depend primarily on exploiting redundancies, particularly frame to frame.

Beyond the previously mentioned common encoding structure lies a galaxy of so-called nonlinear or adaptive techniques wherein the reconstruction is a much more complicated function of the channel sequence. A good example is variable step size DM. Several algorithms have been suggested whereby the decoder senses changes in the parameters

of the source, from the channel sequence, and accordingly varies the step size. Such adaptive reconstruction rules are also candidates for improvement by weighted noise encoding. However, the additional complexity required of the encoder is sometimes significant, and scant results have indicated that much larger encoding block sizes are required because of the nonlinear behavior of the reconstruction.

# Chapter II

## 2.1 Overview of Encoder Philosophy

The basic principle of the encoder design is to minimize with each block decision an estimator of the weighted noise power, $\hat{d}_{WN}$, which is derived from the finite history of past errors

$$\underline{q}_{-M}^{-1} = \underline{s}_{-M}^{-1} - \hat{\underline{s}}_{-M}^{-1} \qquad (2.1.1)$$

in conjunction with the N future errors $\underline{q}_0^{N-1}$ forthcoming as a result of that N-fold decision. Assuming an alphabet size of K, ie, a K-ary system, there are in general $K^N$ possibly distinct sequences $\underline{q}_{-M}^{N-1}$ of past and future errors concatenated. In effect, the estimator is individually computed for all $K^N$ error patterns, and that digital sequence $\underline{c}_0^{N-1}$ giving rise to the error pattern which is estimated to produce the least weighted noise is chosen for transmission.

This assumes three basic attributes of the encoder. First, the encoder must have knowledge of N new source samples prior to encoding the next block. In practice this requires an absolute delay of $(N-1)\tau$ seconds, along with storage for N-1 source values. Then, there must be additional storage for M values corresponding to the immediately previous errors. Lastly, the encoder must be able to generate the error sequence. This is accomplished with a local decoder identical to the distant one which provides at

the encoder a replica of the reconstruction sequence for subtraction from the source sequence. As stated in Chapter 1, the decoder, ie, the mapping R, is fixed in an ad-hoc fashion. No attempt is made to optimize it.

Now it is patently ridiculous to generate (from the local decoder) each possible $\underline{\hat{s}}_0^{N-1}$, then form all $\underline{q}_{-M}^{N-1}$, and compute the weighted noise estimator $K^N$ times at every encoding. This would be prohibitively complex and time consuming job even for $K = 2$ and small N, since any reasonable estimator involves many computations.

Consider the N-dimensional space of real valued sources samples which are to be block encoded. Minimizing $\hat{d}_{WN}\left(\underline{q}_{-M}^{N-1}\right)$ induces a partition of this space into $K^N$ regions, each region being identified with an optimum choice of $\underline{c}_0^{N-1}$. If $\underline{s}_0^{N-1}$ falls into the $\nu$-th region, then digital sequence $\underline{c}_{0,\nu}^{N-1}$ leads to the least $\hat{d}_{WN}$ for that N-vector of source samples. The encoding complexity problem is partially overcome by finding a suitable $\hat{d}_{WN}$ such that the partitioning of the source sample space results in simple regions with easily definable boundaries. Encoding therefore reduces to ascertaining in which one of $K^N$ regions the given source sequence vector lies, and if the region boundaries are sufficiently simple, then the process is amenable to well-known logical design techniques.

## 2.2 Weighted Noise Power Estimator

The finite length record $\underset{\sim}{q}_M^{N-1}$ is considered to have been extracted from a sample function of a wide sense stationary stochastic sequence. The desired indicator of weighted noise power is obtained by first estimating the spectral density function of the sequence, and then carrying out the integral of that estimate weighted by $W(\omega)$. The interpretation of spectral density measurement by way of window function, as given in Blackman and Tukey[9], is followed here, and the so-called indirect method is used whereby the autocorrelation is sought first.

From now on it will be convenient to use matrix representation so that $\underset{\sim}{q}_M^{N-1}$ will correspond to a (N+M) dimensional vector $\vec{Q}$ with components re-indexed $\{Q_1 \ldots, Q_L\}$, where L = M + N. First, the apparent autocorrelation of this finite length record is obtained by summing over lagged products

$$\hat{R}_{qq}(n\tau) = \frac{1}{L - |n|} \sum_{h=1}^{L-|n|} Q_h Q_{h+|n|} \quad , \quad |n| \leq L - 1, \tag{2.2.1}$$

where the largest lag time for which data is available is $(L-1)\tau$, and only one product contributes to this term.

Next, for reasons to become clear shortly, the triangular shaped lag window

$$V(n\tau) = \left\{ \begin{array}{ll} 1 - \frac{|n|}{L} & , \quad n = 0, \pm 1, \ldots, \pm(L-1) \\ 0 & , \quad |n| \geq L \end{array} \right\} \tag{2.2.2}$$

is applied to the observed autocorrelation. Upon Fourier series expansion in the windowed autocorrelation values one has the modified spectral density estimate

$$\hat{\Omega}_q(\omega) = \sum_{n=-\infty}^{\infty} V(n\tau)\hat{R}_{qq}(n\tau)e^{i\omega n\tau}, \qquad (2.2.3)$$

and so

$$\hat{\Omega}_q(\omega) = \frac{1}{L} \sum_{n=-(L-1)}^{L-1} \sum_{j=1}^{L-|n|} Q_j Q_{j+|n|} e^{i\omega n\tau}. \qquad (2.2.4)$$

A little shuffling brings this into

$$\hat{\Omega}_q(\omega) = \frac{1}{L} \sum_{j=1}^{L} Q_j^2 + \frac{2}{L} \sum_{n=1}^{L-1} \sum_{j=n+1}^{L} Q_n Q_j \cos\ \omega\tau(n-j) \qquad (2.2.5)$$

which is recognized as a quadratic form in the error vector,

$$\Omega_q(\omega) = \frac{1}{L}\, \vec{Q}^T X \vec{Q} \qquad (2.2.6)$$

and the associated symmetric transformation matrix is given by

$$X_{\xi\eta} = \cos(\xi-\eta)\omega\tau. \qquad (2.2.7)$$

The weighted noise power estimate is then

$$\hat{d}_{WN} = \int_{-\pi/\tau}^{\pi/\tau} \left(\frac{1}{L}\, \vec{Q}^T X \vec{Q}\right) W(\omega) d\ , \qquad (2.2.8)$$

and finally

$$\hat{d}_{WN} = \frac{1}{L}\, \vec{Q}^T B \vec{Q} \qquad (2.2.9)$$

where B is obtained from X by integrating termwise; thus

$$B_{\xi\eta} = \int_{-\pi/\tau}^{\pi/\tau} W(\omega)\cos(\xi-\eta)\omega\tau \, d\omega. \qquad (2.2.10)$$

Since the elements of B are functions only of the unsigned difference of the indices, let

$$b(\xi-\eta) \triangleq B_{\xi\eta} = B_{\eta\xi}. \qquad (2.2.11)$$

A simple and intuitively satisfying relationship exists connecting $\hat{d}_{WN}$ and the true spectral density integrated against W. By definition,

$$E\{\hat{R}_{qq}(n\tau)\} = R_{qq}(n\tau), \qquad (2.2.12)$$

the true autocorrelation. Then using the linearity of the expectation operator,

$$E\{\hat{\Omega}_q(\omega)\} = \sum_{n=-\infty}^{\infty} R_{qq}(n\tau)V(n\tau)e^{i\omega n\tau}. \qquad (2.2.13)$$

A basic property of Fourier analysis allows this to be rewritten as

$$E\{\hat{\Omega}_q(\omega)\} = \left[\sum_{n=-\infty}^{\infty} R_{qq}(n\tau)e^{i\omega n\tau}\right] * \left[\sum_{n=-\infty}^{\infty} V(n\tau)e^{i\omega n\tau}\right] \qquad (2.2.14)$$

where the asterisk denotes convolution. Thus the ensemble average (2.2.13) is the true spectrum convolved with an aliased spectral window $V(\omega)$ given by

$$V(\omega) = \sum_{n=-L}^{L} \left(1 - \frac{|n|}{L}\right) e^{i\omega n\tau} = L\tau \sum_{n=-\infty}^{\infty} \frac{\sin^2\left(\frac{\omega L\tau}{2} - \frac{2\pi n}{\tau}\right)}{\left(\frac{\omega L\tau}{2} - \frac{2\pi n}{\tau}\right)^2}. \tag{2.2.15}$$

As noted in [9], the aliasing of the spectral window is negligible when more than a few, eg, 12, data points are used. So for $L \geq 20$, $V(\omega)$ can be considered to be merely

$$V(\omega) \approx L\tau \frac{\sin^2 \frac{\omega L\tau}{2}}{\left(\frac{\omega L\tau}{2}\right)^2} \tag{2.2.16}$$

in the central band, for example.

Making further use of the linearity, the above steps may be drawn together to get

$$E\{\hat{d}_{WN}\} = E\left\{\int_{-\pi/\tau}^{\pi/\tau} \hat{\Omega}_q(\omega)W(\omega)d\omega\right\} = \int_{-\pi/\tau}^{\pi/\tau} E\{\hat{\Omega}_q(\omega)\}W(\omega)d\omega, \tag{2.2.17}$$

so that

$$E\{\hat{d}_{WN}\} = \int_{-\pi/\tau}^{\pi/\tau} \left(\Omega_q(\omega) * V(\omega)\right)W(\omega)d\omega. \tag{2.2.18}$$

Since convolution and integration commute, an equivalent, yet more useful interpretation, is

$$E\{\hat{d}_{WN}\} = \int_{-\pi/\tau}^{\pi/\tau} \Omega_q(\omega)\left(W(\omega) * V(\omega)\right)d\omega. \tag{2.2.19}$$

Thus $\hat{d}_{WN}$ is an unbiased estimate of the true spectral density integrated over weighting function which has been modified by convolution with the spectral window (the window having resulted from the finite length of the data record).

It is now evident that the triangular lag window was used to obtain a B matrix whose elements are given directly by the Fourier series coefficients of W. This has not only the benefit of simplicity, but ensures that $\vec{Q}^T B \vec{Q} \geq 0$. Indeed, since its elements are the trigonometric moments of the nonnegative W, B is a Toeplitz matrix and the corresponding quadratic form (2.2.9) is a Toeplitz form, and therefore positive definite[†][10].

Furthermore, it is a-fortiori the natural extension of a quadratic cost into higher dimensions, where an interesting special case links it back to MSE. Recall from section 1.2 that MSE is equivalently the noise spectral density integrated over the entire baseband. Then to encode via that criterion one would merely set W to unity everywhere. The computations in this case for the B coefficients are simply

$$B_{\xi\eta} = \int_{-\pi/\tau}^{\pi/\tau} 1 \cdot \cos(\xi-\eta)\omega\tau \, d\omega = \frac{2\pi}{\tau} \delta_{\xi\eta}, \quad (2.2.20)$$

---

† Actually, the weaker result $E\{\hat{d}_{WN}\} \geq 0$ follows immediately from (2.2.19), partly because W is nonnegative, but mainly because the triangular V has a nonnegative transform.

with $\delta$ the Kronecker delta. Thus B becomes the unit (identity) matrix, whence

$$\hat{d}_{WN} = \frac{1}{L} \vec{Q}^T \vec{Q} = \frac{1}{L} \sum_{i=1}^{L} q_i^2, \qquad (2.2.21)$$

which clearly estimates MSE.

To conclude this part, several very desirable properties are embodied in (2.2.9) as a cost function. While it may not be the minimum variance estimator of $d_{WN}$, it is an unbiased estimate of an easily described quantity that is in most cases a negligibly small perturbation of the desired parameter. As will be seen in the following section, its form invites the application of matrix algebra methods to quite simply derive and implement the encoding region boundaries; a task which might prove impossible with another cost function of greater complexity.

## 2.3 Encoding Regions and Boundaries

The encoding rule which stems from the minimization of the vector cost function (2.2.9) is derived. First, an analytical development is made, and then the steps are examined within the more enlightening geometric context. Since only the future errors $\underline{q}_o^{N-1}$ are being controlled by virtue of the choice for $\underline{\hat{s}}_o^{N-1}$, the past errors having already been established by previous encodings, the roles played by $\underline{q}_o^{N-1}$ and $\underline{q}_{-M}^{-1}$ in the process are different. Accordingly, that distinction is maintained by letting $\vec{Q}_p$ and $\vec{Q}_f$ denote the M and N dimensional vector equivalents of the past and future error sequences, respectively.

Using partitioned matrix notation, the weighted noise estimator may be expressed as (dropping the unimportant constant 1/L)

$$\hat{d}_{WN} = \left[\vec{Q}_f^T \;\vdots\; \vec{Q}_p^T\right] \left[\begin{array}{c:c} B_N & \beta \\ \hdashline \beta^T & B_M \end{array}\right] \left[\begin{array}{c} \vec{Q}_f \\ \hdashline \vec{Q}_p \end{array}\right] \qquad (2.3.1)$$

where $\beta^T$ is the transpose of $\beta$, and the principal submatrices of B are subscripted to signify their order. Carrying out the indicated products gives

$$\hat{d}_{WN} = \vec{Q}_f^T B_N \vec{Q}_f + \vec{Q}_f^T \beta \vec{Q}_p + \vec{Q}_p^T \beta^T \vec{Q}_f + \vec{Q}_p^T B_M \vec{Q}_p \qquad (2.3.2)$$

The last term involves only the past errors, and so acts as a constant, say $\lambda$. The middle two terms, which represent

cross products between past and future errors, are easily seen to be transposes, and are therefore equal. Then the above becomes

$$\hat{d}_{WN} = \lambda + \vec{Q}_f^T B_N \vec{Q}_f + 2\vec{Q}_f^T \beta \vec{Q}_p \qquad (2.3.3)$$

which may be placed in better form by a translation of coordinates analogous to completing the square. The intermediate step is, after adding a term which contains only past errors,

$$\hat{d}_{WN} = \lambda' + \vec{Q}_f^T B_N \vec{Q}_f + 2\vec{Q}_f^T \left(B_N B_N^{-1}\right)\beta \vec{Q}_p + \left(B_N^{-1}\beta \vec{Q}_p\right)^T B_N \left(B_N^{-1}\beta Q_p\right) \qquad (2.3.4)$$

and this reduces to

$$\hat{d}_{WN} = \lambda' + \left(\vec{Q}_f + B_N^{-1}\beta \vec{Q}_p\right)^T B_N \left(\vec{Q}_f + B_N^{-1}\beta \vec{Q}_p\right) \qquad (2.3.5)$$

Once again, $\lambda'$ is a function only of $\vec{Q}_p$ and therefore minimization may be carried out with respect to the simpler quadratic form

$$\hat{d}'_{WN} = (\vec{Q}_f + \vec{\Psi})^T B_N (\vec{Q}_f + \vec{\Psi}) \qquad (2.3.6)$$

where

$$\vec{\Psi} \triangleq B_N^{-1}\beta \vec{Q}_p \qquad (2.3.7)$$

is an N-vector which depends linearly on the past errors, and may be thought of as a biasing of the future error vector in the form (2.3.2).

Since B is positive definite, its principal submatrices, in particular $B_N$, are also positive definite. Therefore an immediate consequence of (2.3.6) is that the future error which minimizes $\hat{d}_{WN}$ is given by $\vec{Q}_f + \vec{\Psi} = 0$. Furthermore, $\lambda'$ can be thought of as that portion of the weighted noise estimate coming from the past errors which cannot be compensated by adjustment of $\underline{q}_o^{N-1}$. Of course, the opportunity to set $\vec{Q}_f = -\vec{\Psi}$, even approximately, in any given encoding will be a rare event. The idea is simply to come as close as possible by selecting the minimum of (2.3.6) over all error sequences available.

The above can be stated in geometric terms, and this provides insight about the way in which decisions affect the noise pattern, as well as helps to make the algorithm intuitively appealing. Consider an L-dimensional real space which contains $K^N$ fixed vectors corresponding to the various outcomes

$$\left\{\begin{matrix}\hat{s}^{N-1} \\ \underline{s}_M\end{matrix}\right\}_\nu \Rightarrow \vec{\hat{S}}_\nu = \begin{bmatrix}\vec{\hat{S}}_{f,\nu} \\ \hline \vec{\hat{S}}_{p,\nu}\end{bmatrix}, \quad \nu = 1,\ldots,K^N \qquad (2.3.8)$$

The source sequence is also an L vector $\vec{S}$ in this space, but may be considered as only ranging over an N-dimensional subspace of it. Encoding amounts to finding the reconstruction vector $\vec{\hat{S}}_f$ which lies closest to the biased source vector, $\vec{S}_f + \vec{\Psi}$.

However, distance in this space is not given by the usual (Euclidean) norm of the vector difference, but is instead the nonisotropic measure

$$d(\vec{U},\vec{V}) = \sqrt{(\vec{U}-\vec{V})^T B(\vec{U}-\vec{V})} \qquad (2.3.9)$$

for two vectors $\vec{U}$ and $\vec{V}$. It is clear that $\hat{d}_{WN}$ and $d(\vec{S}_f+\vec{\Psi},\hat{\vec{S}}_f)$ are simultaneously minimized; the positive definite form gives positive distances, and square root is monotone increasing.

The steps carrying (2.3.1) into (2.3.6) are seen to be equivalent geometrically to a projection of all points $\vec{S}$, $\hat{\vec{S}}$ onto the subspace. Indeed, one could determine the encoding regions in the entire L-space, and then project them to the N-space, which has effectively been done if (2.3.6) is used. It is sufficient, therefore, to consider source sample spaces with dimensionality N, which shall be called the block size.

A future source vector $\vec{S}_f$ will be in the $\nu$-th encoding region $\Theta_\nu$ if and only if $\vec{S}_f+\vec{\Psi}$ is closest to reconstruction vector $\hat{\vec{S}}_{f,\nu}$. In set notation,

$$\Theta_\nu = \bigcap_{\substack{\xi=1 \\ \xi\neq\nu}}^{K^N} \{\vec{S}_f \mid d(\vec{S}_f+\vec{\Psi},\hat{\vec{S}}_{f,\nu}) < d(\vec{S}_f+\vec{\Psi},\hat{\vec{S}}_{f,\xi})\} \qquad (2.3.10)$$

Using (2.3.6) it is a simple matter to determine the half spaces which intersect to form the projection of $\Theta_\nu$ onto the subspace. Initially, suppose that $d(\vec{S}_f+\vec{\Psi},\hat{\vec{S}}_{f,\nu}) < d(\vec{S}_f+\vec{\Psi},\hat{\vec{S}}_{f,\xi})$ whence

$$(\vec{S}_f-\hat{\vec{S}}_{f,\nu}+\vec{\Psi})^T B_N(\vec{S}_f-\hat{\vec{S}}_{f,\nu}+\vec{\Psi}) < (\vec{S}_f-\hat{\vec{S}}_{f,\xi}+\vec{\Psi})^T B_N(\vec{S}_f-\hat{\vec{S}}_{f,\xi}+\vec{\Psi}) \tag{2.3.11}$$

Now expand both sides, and cancel common terms $\vec{S}_f^T B_N \vec{S}_f$ to get after factoring,

$$2(\hat{\vec{S}}_{f,\nu}-\hat{\vec{S}}_{f,\xi})^T B_N \vec{S}_f + (\vec{\Psi}-\hat{\vec{S}}_{f,\xi})^T B_N(\vec{\Psi}-\hat{\vec{S}}_{f,\xi}) - (\vec{\Psi}-\hat{\vec{S}}_{f,\nu})^T B_N(\vec{\Psi}-\hat{\vec{S}}_{f,\nu}) \geq 0. \tag{2.3.12}$$

Then use the identity (analogous to $x^2 - y^2 = (x-y)(x+y)$) for hermitian B,

$$\vec{X}^T B\vec{X} - \vec{Y}^T B\vec{Y} = (\vec{X}-\vec{Y})^T B(\vec{X}+\vec{Y})$$

to arrive at the result

$$d(\vec{S},\vec{S}_\nu) < d(\vec{S},\vec{S}_\xi) \Rightarrow (\hat{\vec{S}}_{f,\nu}-\hat{\vec{S}}_{f,\xi})^T B_N\left(\vec{S}_f + \vec{\Psi} - \tfrac{1}{2}[\hat{\vec{S}}_{f,\nu}+\hat{\vec{S}}_{f,\xi}]\right) > 0 \tag{2.3.13}$$

Replacing the inequality above with an equality gives the equation for the boundary separating the two volumes which correspond to points closer to $\hat{\vec{S}}_{f,\nu}$ than $\hat{\vec{S}}_{f,\xi}$, and vice-versa. It is, by definition, the locus of points, in this case a hyper-surface, which are equidistant from the two reconstruction vectors.

For convenience, define two new (N-dimensional) vectors

$$\left.\begin{aligned} \vec{\Delta}_{\nu\xi} &= \hat{\vec{S}}_{f,\nu} - \hat{\vec{S}}_{f,\xi} \\ \vec{\Sigma}_{\nu\xi} &= \tfrac{1}{2}\left[\hat{\vec{S}}_{f,\nu}+\hat{\vec{S}}_{f,\xi}\right] \end{aligned}\right\} \tag{2.3.14}$$

in order to rearrange (2.3.13) more usefully as

$$\vec{\Delta}^T_{\nu\xi} B_N \vec{S}_f > \vec{\Delta}^T_{\nu\xi} \left( B_N \vec{\Sigma}_{\nu\xi} - \beta \vec{Q}_p \right) \qquad (2.3.15)$$

This avoids inverting $B_N$, and at the same time emphasizes several interesting aspects of the boundary. First, it is a hyperplane, since the equation is a linear relation in the components of $\vec{S}_f$. The orientation of the hyperplane is a function of $\vec{\Delta}_{\nu\xi}$, which is its normal. The boundary is translated back and forth according to the past errors $\vec{Q}_p$ which enter linearly into the constant on the right hand side.

Most importantly, since the decision between candidate outcomes $\hat{\vec{S}}_\nu$ and $\hat{\vec{S}}_\xi$ involves only a single threshold, implementation of the logic to place $\vec{S}_f$ into the proper encoding region is greatly facilitated, and even results in practicable hardware realizations for reasonably simple decoders and moderate block lengths.

## 2.4 Connection with Channel Detection

The minimization of a positive definite quadratic form in the vector difference between a given point and several fixed loci in an N-space is also an important vector channel detection problem. When the channel noise is a non-white Gaussian process with, perhaps, non-zero mean, the detector implementation problem and the minimum $\hat{d}_{WN}$ source encoding problem are formally identical. In the following description, the symbols used will be those of the analogous source encoding quantities.

The channel transmitter communicates one of $K^N$ messages every $N\tau$ seconds by sending a corresponding N-vector $\vec{\hat{S}}_{f,\nu}$. The decoder receives a corrupted version

$$\vec{S}_f = \vec{\hat{S}}_{f,\nu} + \vec{Q}_f \qquad (2.4.1)$$

where $\vec{Q}_f$ is a noise vector whose N components are jointly Gaussian random variables, with means

$$E\{\vec{Q}_f\} = -\vec{\Psi} \qquad (2.4.2)$$

and (positive definite) covariances

$$E\{(\vec{Q}_f+\vec{\Psi})(\vec{Q}_f+\vec{\Psi})^T\} = B_N^{-1} \qquad (2.4.3)$$

which is not a diagonal matrix* since the noise components are correlated (colored noise). The decoder attempts to identify the message by choosing that $\hat{\vec{S}}_{f,\nu}$ for which the conditional probability (for equally likely messages)

$$\Pr\{\vec{S}_{f,\nu}|\vec{S}_f\} = \sqrt{\frac{\det B_N}{(2\pi)^N}}\, e^{-\frac{1}{2}(\vec{S}_f-\hat{\vec{S}}_{f,\nu}+\vec{\Psi})^T B_N(\vec{S}_f-\hat{\vec{S}}_{f,\nu}+\vec{\Psi})} \tag{2.4.4}$$

is greatest over all $\nu$ in the signal vector set. This is clearly maximized when the exponent is minimum, ie,

$$d(\vec{S}_f+\vec{\Psi},\ \hat{\vec{S}}_{f,\nu})$$

is least. Hence the identification with the present source encoding problem.

The author is unaware of any techniques from channel detection theory for implementing the minimum distance search, other than straightforward computation. Obviously, if some method were available it would be directly applicable here. Conversely, the sequential encoding algorithm outlined in Chapter 4 may find application in detecting block orthogonal signal sets in colored Gaussian noise.

---

* In the stationary, white noise case $B_N^{-1}$ is of the form $\sigma^2 I_N$, where $I_N$ is the identity matrix, and this corresponds in the the source encoding case to no frequency weighting, ie, mean square error encoding.

# Chapter III

## 3.1 Block Length One Encoders

The simplest form for the encoder is obtained when the block length shrinks to one. Naturally, it then bears the greatest resemblance to its standard counterparts; however the larger block length encoders, to be treated in the following chapter, depart radically in form from the conventional single sample at a time types. To simplify terminology, the N = 1 encoders will be called block-1, the N = 2 called block-2, etc. Also, the "block" designation implies the use of the minimum $\hat{d}_{WN}$ decision boundaries, as opposed to standard decision rules (which always pertain only to sample by sample processing).

When N = 1, the encoding space becomes the real line, and $\vec{\Delta}_{\nu\xi}$, $\vec{\Sigma}_{\nu\xi}$, $B_N$ reduce to scalars, thus

$$\left.\begin{aligned} \Delta_{\nu\xi} &= \hat{s}_{0,\nu} - \hat{s}_{0,\xi} \\ \Sigma_{\nu\xi} &= \frac{1}{2}\left(\hat{s}_{0,\nu} + \hat{s}_{0,\xi}\right) \\ B_N &= b(0) \end{aligned}\right\} \qquad (3.1.1)$$

Then $\beta$ becomes a row vector, so that $\beta\vec{Q}_p$ is an inner product, and

$$B_N^{-1}\beta Q_p = \sum_{j=1}^{M} q_{-j}\,\frac{b(j)}{b(0)} \equiv \Psi_0 \qquad (3.1.2)$$

The boundary equation for block-1 encoding (2.3.15) therefore becomes, after cancelling common terms $\Delta_{\nu\xi}^{T} B_N$

$$s_0 > \frac{1}{2} (\hat{s}_{0,\nu} + \hat{s}_{0,\xi}) - \Psi_0 \qquad (3.1.3)$$

for choosing $\hat{s}_{0,\nu}$ over $\hat{s}_{0,\xi}$ provided that $\hat{s}_{0,\nu} > \hat{s}_{0,\xi}$. A more illustrative rearrangement of the threshold is

$$s_0 + \Psi_0 > \frac{1}{2} (\hat{s}_{0,\nu} + \hat{s}_{0,\xi}) \qquad (3.1.4)$$

This result may be easily visualized, as in Figure 3.1, by graphing the cost $\hat{d}_{WN}$ as a function of $s_0$, with $\hat{s}_0$ as a parameter. With $\vec{Q}_p$ fixed, it is clear that the graph of $\hat{d}_{WN}$ vs. $q_0$ is a parabola, one which opens upward since B is positive definite. From (3.1.2) the minimum of the curve is at

$$q_0(\min) = -\Psi_0 \qquad (3.1.5)$$

and since $q_0 = s_0 - \hat{s}_0$, the desired function is just the same quadratic centered about the point

$$s_0 \text{opt} = \hat{s}_0 - \Psi_0 \qquad (3.1.6)$$

In the figure, a separate cost curve is drawn for each $\hat{s}_{0,\xi}$, $\xi = 1,\ldots,K$, and the lowest cost at each point on the $s_0$ axis determines encoding regions, which are line segments in this one dimensional case. Because the conditional cost is symmetric and increasing, the intersections of the functions, which are the region boundaries, occur at the midpoints between respective centers $s_0(\text{opt})$. The

conclusion to be drawn, either from Equation (3.1.4) or by inspection of the figure, is that block-1 encoding is identical with any single digit, minimum distance rule once the input sample has been preadjusted by the addition of $\Psi_o$.

## 3.2 Application to PCM

Uniform PCM, being a minimum distance decision rule, is adapted to Block-1 encoding by slightly increasing the sampling rate, and adding $\Psi_0$ to the input. The encoder is shown in block diagram form in Figure 3.2. The amount by which the sampling rate may be increased is dictated by the system parameters, but if only a very small percentage is allowed, the improvement seen may be small. This is even more the situation when the original number of quantizing levels, equivalently the number of transmission bits per sample, is large.

First, if the percentage increase in sampling rate over the Nyquist rate is small, the corridor of frequencies just above the highest signal component into which the quantizing noise is to be concentrated is a correspondingly small fraction of the baseband. In this case the exact shape of the spectral window function, resulting from the use of a finite number of points to estimate the noise spectrum, becomes important. The problem is that the effective weighting function $W(\omega)*V(\omega)$(cf.2.2.19 ) cannot decrease to near zero over a significant region of the unused corridor. Consequently, the noise will not be shifted to that zone to the degree desired.

Two things may be done to improve the effective weighting, and both amount to modifying the lag window V. By taking more points, ie, increasing M + N, the entire

spectral window is compressed on the frequency scale. Then, with a more sophisticated window itself the side-lobe structure of $V(\omega)$ can be improved. This, of course, would merely result in a different set of coefficients for the feedback filter, and alter the shape of the encoding regions for $N \geq 2$.

It is felt that improving PCM by noise weighting approaches, such as Block-1 encoding, is perhaps not as fruitful as with the higher sampling rate systems. For example, given an 8 bit PCM system, a 25 percent increase in sampling rate would allow 2 additional bits for transmission, which immediately improves performance about 12 dB. Furthermore, a narrow corridor above the signal band which contains very high level noise imposes severe requirements on the decoder output filter to reject that noise while passing the signal.

The above reasoning partly explains the lack of interest in noise feedback endocing, since its introduction around 1960, because only low sampling rates were considered. Also, PCM is inherently a complicated system, and adding the computational burden of Block-1 encoding to it is felt to be less effective than using all that complexity to implement a good adaptive system.

## 3.3 Relationship of Spang's Noise Feedback Encoder to Block-I PCM

The schematic diagrams of Block-I PCM, and the Spang and Schultheiss (S-S) noise feedback encoder shown in Figure 3.3, while quite similar at first glance, do display a subtle difference. First, the feedback coefficients, or tap weights, are not quite alike. Most significantly, the two designs spring from entirely different origins. In order to discuss the differences, an abbreviated derivation of their encoder is given below. The z-transform theory of sampled data systems is used to short circuit their more involved analysis, which was based on autocorrelations and joint probability distributions.

The crux of the S-S analysis is the assumption that the uniform, multilevel quantizer behaves as a unity gain amplifier with an internal white, random noise source $n$ with variance $\delta^2/12$, where $\delta$ is the interlevel spacing. Referring to the block diagram in Figure 3.4a, the quantizer is replaced by a device which merely adds to its input this uncorrelated noise $n(z)$, which is also assumed to be statistically independent of the input, $s(z)$.

By inspection, the input to the feedback network is $-n(z)$. Representing the feedback by its transfer function

$$G(z) = H_1 z^{-1} + \ldots + H_M z^{-M} \qquad (3.3.1)$$

the input to the quantizer is the sum $\delta(z) - n(z)G(z)$. Now employing the quantizer model, the system output is written directly

$$\hat{\delta}(z) = \delta(z) + n(z)[1 - G(z)] \qquad (3.3.2)$$

The feedback does not affect the signal, but the spectral density function of the internal quantizer noise is modified by the factor

$$H(\omega) = \left|1 - G(z)\right|^2_{z=\exp(i\omega\tau)} = \left(\sum_{r=0}^{M} H_r e^{i\omega r\tau}\right)\left(\sum_{r=0}^{M} H_r e^{-i\omega r\tau}\right) \qquad (3.3.3)$$

where $H_o$ is defined to equal -1.

An equivalent model of the encoding system, then, is as shown in Figure 3.4b where the signal is corrupted by originally flat spectrum independent quantizing noise which has been passed through a feed-forward type of digital filter. Even though the quantizing noise is fed back, the filtering clearly is nonrecursive, and $1 - G(z)$ contains at most only finite transmission zeros.

The noise filter transfer function can be expressed as a quadratic form in the M + 1 dimensional vector of coefficients $H_o, H_1, \ldots, H_M$ as

$$H(\omega) = \frac{\delta^2}{12} \vec{H}^T X \vec{H} \qquad (3.3.4)$$

where X is the matrix (2.2.7), and so the frequency weighted noise power is proportional to $H^T BH$ (c.f. 2.2.8 ). Recall that $\hat{d}_{WN}$, the weighted noise power estimate, is this same form, only the argument is the error vector $\vec{Q}$.

Choosing the filter coefficients is not so simple as minimizing the quadratic form in $\vec{H}$ over the constraint plane $H_o = -1$. The inclusion of the feedback significantly increases the variance of quantizer input above that of the signal itself. Therefore to maintain the probability of saturation, or of exceeding the range of the quantizer, below that value which would invalidate the independent additive white noise quantizer model, the number of quantizer steps must be increased accordingly. This would, on the other hand, make for an unfair comparison to encoding without feedback, since the additional quantizer levels could otherwise be distributed so as to reduce $\delta$, and thereby decrease the noise.

S-S did the following the reconcile this factor when optimizing the feedback coefficients. The worst case is when the previous M errors are of largest possible magnitude $\delta/2$ (no saturation) and signed so that the filter output is

$$\pm \frac{\delta}{2} \sum_{j=1}^{M} |H_j|.$$

A noise feedback encoder with, say, a 2k level quantizer is using in the worst case $\sum_{j=1}^{M} |H_j|$ additional levels to

guard against overload. Consequently, an equivalent standard encoder with the same number of levels would have $\delta$ reduced by the factor $2k - \sum_{j=1}^{M} |H_j|$. Using that reasoning, S-S defined an improvement criterion based on the ratio of in-band noise power with error feedback to the noise power without it, keeping the sampling rate and alphabet size fixed. Except for some multiplicative constants in front, this led to the positive quantity

$$\frac{\vec{H}^T B \vec{H}}{\left(1 - \frac{1}{2k} \sum_{j=1}^{M} |H_j|\right)^2}$$

which is to be minimized with respect to the coefficient vector $\vec{H}$, constrained to lie on the plane $H_0 = -1$.

This appeared to be a prohibitively difficult problem for analytical solution, and S-S resorted to a computer search for the optimum $\vec{H}$. However, in their calculations they stressed the limiting case $k \uparrow \infty$, for which it will be seen an analytical solution is readily obtained. It goes, informally, as follows. In the limit of large k, the denominator becomes unity, and only the numerator quadratic is left to be minimized. It suffices that

$$\left(\vec{H}_{opt}^T \vec{H}_{opt}\right)^{1/2} = o(K)$$

which is quite reasonable. Proceeding much the same as in

section 2.3, $\vec{H}$ is dichotomized into the free elements $H_1,\ldots,$ $H_M$, and the fixed $H_0$, which leads to the partitioned formulation

$$\vec{H}^T B \vec{H} = \left[-1 \;\vdots\; \vec{H}_M^T\right] \left[\begin{array}{c|c} 1 & \vec{\beta} \\ \hline \vec{\beta}^T & B_M \end{array}\right] \left[\begin{array}{c} -1 \\ \hline \vec{H}_M \end{array}\right] \tag{3.3.5}$$

where

$$\vec{\beta}^T = \begin{bmatrix} b(1) \\ b(2) \\ \vdots \\ b(M) \end{bmatrix} \quad , \quad \vec{H}_M \triangleq \begin{bmatrix} H_1 \\ \vdots \\ H_M \end{bmatrix} . \tag{3.3.6}$$

After expanding, and combining the equal cross terms, the numerator becomes

$$\vec{H}^T B \vec{H} = 1 - 2\vec{\beta}\vec{H}_M + \vec{H}_M^T B_M \vec{H}_M \; . \tag{3.3.7}$$

Upon completing the matrix product square by adding and subtracting

$$\vec{\beta} B_M^{-1} \vec{\beta}^T \equiv \left(B_M^{-1}\vec{\beta}^T\right)^T B_M \left(B_M^{-1}\vec{\beta}^T\right) \tag{3.3.8}$$

one gets

$$\vec{H}^T B \vec{H} = \left(\vec{H}_M - B_M^{-1}\vec{\beta}^T\right)^T B_M \left(\vec{H}_M - B_M^{-1}\vec{\beta}^T\right) \tag{3.3.9}$$

$$+ \text{ terms not depending on } \vec{H}_M .$$

Once again, since $B_M$ is positive definite, the minimum is achieved by setting

$$\vec{H}_M opt = B_M^{-1}\vec{\beta}^T, \qquad (3.3.10)$$

but this is a result not mentioned in S-S.

Now analyze the block-1 encoder using the same linearizing assumption for the quantizer. Notice that here the feedback is derived from the overall input-output error, as contrasted with quantizer error feedback in the S-S version. One sees from Figure 3.2 that the filter input is $s(z) - \hat{s}(z)$ and so

$$\hat{s}(z) = s(z) + n(z) + [s(z) - \hat{s}(z)]B(z) \qquad (3.3.11)$$

where

$$B(z) = b(1)z^{-1} + b(2)z^{-2} + \ldots + b(M)z^{-M}. \qquad (3.3.12)$$

Therefore the result is

$$\hat{s}(z) = s(z) + n(z)[1 + B(z)]^{-1} \qquad (3.3.13)$$

In this case the feedback acts as a recursive filter on the white noise source in the quantizer, as evidenced by the expansion for the filter transfer function

$$[1 + B(z)]^{-1} = 1 - B(z) + B^2(z) - B^3(z) + \ldots \qquad (3.3.14)$$

which contains terms of all positive orders in $z^{-1}$. Although the optimum vector of feedback gains was easily found for the S-S encoder, a similar solution for this case was not seen.

The problem is that now the polynomial in $\omega$, whose coefficients are related to the b's, appears in the denominator of the integral to be minimized.

Nevertheless, it can be seen that the vector of feedback coefficients which results from the minimum $\hat{d}_{WN}$ analysis, viz. $\vec{\beta}$, is consistent with a small value for the integral, if not the actual solution. $1 + B(z)$ has a low pass response over the range of signal frequencies. Its inverse is therefore a low-band-stop characteristic, which is the proper form to block the white noise over the signal band.

To summarize this section, block-1 PCM is an encoder which is quite similar to the noise feedback design studied in Reference (6). The different way in which the error signal is derived results in feedback filters which are roughly inverses of each other. With regard to the filter coefficients, the S-S values are such as to minimize the response of the filter over the band of interest, with the quantization process itself obscured in the linearizing model. The present technique, on the other hand, concentrates on the actual encoding in an effort to minimize an estimator of the noise, and the coefficients are given directly by that estimate.

Both approaches are seen to have advantages as well as weaknesses, and are in some measure complementary. It is

felt, however, that the present one, considering only single digit encoding, is the more general and flexible. A prime example of this is the application to DM. Two level quantization is a gross violation of the conditions for replacing the quantizer by an independent white noise generator. Yet this is the system which is most improved by encoding to minimize only the in-band noise, Furthermore, the concept of quantizing noise feedback cannot be extended to block encoding, as in naturally done with the noise spectrum estimator approach.

## 3.4 Application to Delta Modulation

The reconstruction mapping for standard, so-called linear single integration, DM is the particularly simple rule

$$\hat{s}_o = \begin{cases} \hat{s}_{-1} + \delta & , \quad c_o = \text{"1"} \\ \hat{s}_{-1} - \delta & , \quad c_o = \text{"-1"} \end{cases}$$

in which $\delta$ is the receiver step size, and the commonly used binary channel symbols "-1", "1" have been introduced. Block-1 DM encoding is now readily obtained from (3.1.4). There are only two alternative reconstruction values, and their average is clearly $\hat{s}_{-1}$, which gives the single encoding boundary

$$s_o + \Psi_o > \hat{s}_{-1} \qquad (3.4.1)$$

for choosing "1" over "-1".

Once again, except for the term involving past errors, $\Psi_o$, that rule is precisely the standard algorithm for this system. Block-1 DM could therefore be implemented as an applique to a standard DM encoder, wherein the past errors are fed back through a transversal filter to add to the input sample ahead of the binary quantizer. This is shown in Figure 3.5 where the dashed line encloses that part which is the standard encoder.

When analyzing this, and the following encoder block diagrams, it is crucial to keep straight the order in which signals are computed and shift register contents are

updated. For example, the feedback shift register cells in Figure 3.5 are labeled with their contents just prior to the $c_0$ encoding, say $t_0^-$. At $t = t_0$, $s_0$ appears on the signal line and $\hat{s}_{-1}$ is still the local decoder output. The value of $c_0$ is decided then, and the local decoder instantaneously updates so that its output is $\hat{s}_0$ at $t_0^+$. Assume that $s_0$ remains at the input just long enough for the difference $q_0 = s_0 - \hat{s}_0$ to be formed in the subtractor, and this feeds to the filter which then updates and recomputes $\Psi_0(t_1)$, a procedure that may occupy the entire time interval $\left(t_0^+, t_1^-\right)$.

## 3.5 Application to a Simple Adaptive DM

The application of Block-1 encoding to a simple nonlinear DM system will now be derived, for illustrative purposes. Abate's scheme [12] is chosen for this example as one which is easily described, and yields a decision rule which is only slightly more complicated than linear Block-1 DM.

To motivate the Abate DM, and indeed most of the so-called instantaneous adaptive encoders, consider that the noise performance is strongly dependent on the step size for a given input. the problem is one of dynamically adjusting $\delta$ to optimize the trade-off between the crudeness of reconstruction ($\delta$ too large) and the likelihood for slope overload ($\delta$ too small). Whereas PCM overloads whenever the input exceeds the largest reconstruction value, DM overloads when the derivative of the input exceeds $\delta/\tau$, the greatest average slope that the reconstruction is able to attain.

When DM undergoes slope overload, the reconstruction becomes a staircase, seeking to catch up to the rapidly slewing input. The channel output is accordingly a sequence of like symbols, eg, "1", "1",... for a positive going slope. A repeating channel symbol, therefore is indicative of slope overload.

With that in mind, the Abate reconstruction rule (with uniformly spaced step sizes) is as follows. Let $\delta$ be some small, basic step size, with the actual admissable $\hat{s}$

transitions being the set $\{\delta, \delta+\gamma, \delta+2\gamma, \ldots, \delta+\gamma_{max}\}$ . Assume the previous transition was made with a nonextreme step size, that is

$$|\hat{s}_{-1} - \hat{s}_{-2}| = \delta + J\gamma \quad , \quad J\gamma \leq \gamma_{max} \tag{3.5.1}$$

Then the next transition will be made with step size $\delta + (J+1)\gamma$ if $c_0 = c_{-1}$, or with step size $\delta + (J-1)\gamma$ if $c_0 \neq c_{-1}$. If already there, the step size remains at the ground state $\delta$ when a reversal occurs. The desired effect is that the step size will tend to hover about a value which reflects the slope content of the signal, thereby causing the system to adapt to it.

The Block-1 Abate DM encoding rule is now obtained by enumerating the reconstruction outcomes

$$\left.\begin{aligned} \hat{s}_0(1) &= \hat{s}_{-1} + \delta + (J+1)\gamma \\ \hat{s}_0(-1) &= \hat{s}_{-1} - [\delta + (J-1)\gamma] \end{aligned}\right\} , \ \hat{s}_{-1} > \hat{s}_{-2} \text{ or } c_{-1} = \text{"-1"} \tag{3.5.2}$$

and

$$\left.\begin{aligned} \hat{s}_0(1) &= \hat{s}_{-1} + \delta + (J-1)\gamma \\ \hat{s}_0(-1) &= \hat{s}_{-1} - [\delta + (J+1)\gamma] \end{aligned}\right\} , \ \hat{s}_{-1} < \hat{s}_{-2} \text{ or } c_{-1} = \text{"0"} \tag{3.5.3}$$

Taking the average reconstruction value, one sees that

$$\tfrac{1}{2}\left(\hat{s}_0(1) + \hat{s}_0(-1)\right) = \hat{s}_{-1} + \gamma \, \mathrm{sgn}(\hat{s}_{-1} - \hat{s}_{-2})$$

which gives the Block-1 Abate DM encoding boundary,

$$s_0 + \Psi_0 = \hat{s}_{-1} + \gamma \mathrm{sgn}(\hat{s}_{-1} - \hat{s}_{-2})$$

This rule is modified slightly if the previous step size was extreme by halving the correction term $\gamma \mathrm{sgn}(\hat{s}_{-1} - \hat{s}_{-2})$.

## 3.6 Relationship of Block-1 DM to a Double Integrator DM

Brolin and Brown[13] considered modifying single integration (standard) DM by including a second integrator-like R-C analog network between the error node (subtractor) and the binary quantizer, as shown in Figure 3.6a. This is not the usual double integration DM system because the endocer is left unchnaged, ie, it consists of a single integrator and a low-pass filter. Suprisingly, this simple addition achieves a definite reduction of the in-band noise over standard DM, by noise spectral shaping.

Later, Brolin[14] identified the operation of the additional circuit as providing a bias to the input, or equivalently to the decision threshold, which makes the error tend to alternate in polarity more frequently. Since the network is in the error path, the bias is a weighted accumulation of past errors; the weighting is determined by the RC discharge (impulse response) and extends into the infinite past. Indeed, it will now be shown that this encoder is intimately related to block-1 DM, and may be considered a primitive precursor to single digit, weighted noise DM encoding.

If in the diagram of Figure 3.5, the summer and subtractor in the signal path to the quantizer are placed in reverse order, the quantizer input remains unaltered; but the two subtractors are now in parallel in the sense that they share the same pair of inputs. Since their inputs are

identical, one of them may be eliminated, in particular the subtractor which feeds the transversal filter, and the filter input is equivalently obtained from the main subtractor. This is illustrated in Figure 3.6b, which reveals that block-1 DM is equivalent to placing a nonrecursive digital filter in the forward path before the quantizer. Once again, though, care must be taken that computations are ordered correctly. The input to the shift register must be gated at $t_n^+$, in order that the proper error voltage is deposited in the first shift register cell.

Since the transfer function of the filter, easily seen to be $1 + \sum_{j=1}^{M} b(j)z^{-j}$, is low pass, the identification with the analog circuit in Figure 3.6a is obvious. In fact, the analog circuit could be realized as a digital filter by making the tap gains equal to the sampled impulse response of the analog network. This requires an infinite number of taps in a nonrecursive filter without approximation. Conversely, an analog realization of block-1 DM may be obtained by synthesizing a network whose impulse response f(t) agrees with b(j) for $t = j\tau$, $j = 0,\ldots,M$, neglecting truncation error.

Based on this research, therefore, it may be said that the second integrator is a step in the right direction, yet the very soft low pass characteristic afforded by such a simple R-C structure (eg, 6 dB/octave slope) is inadequate to

make nearly optimum weighted noise decisions.* This comment will be amplified when, in the discussion of the experimental results in Chapter 5, the effects of various weighting functions, and thus attendant tap gain vectors, are compared.

---

* More precisely, one could determine the effective block-1 $W(\omega)$ corresponding to the analog circuit by reasoning backwards from the impulse response samples, to the equivalent digital filter taps, thence to the W which gives those values. This involves nothing more than a Fourier cosine expansion with coefficients $b(j)/b(0)$.

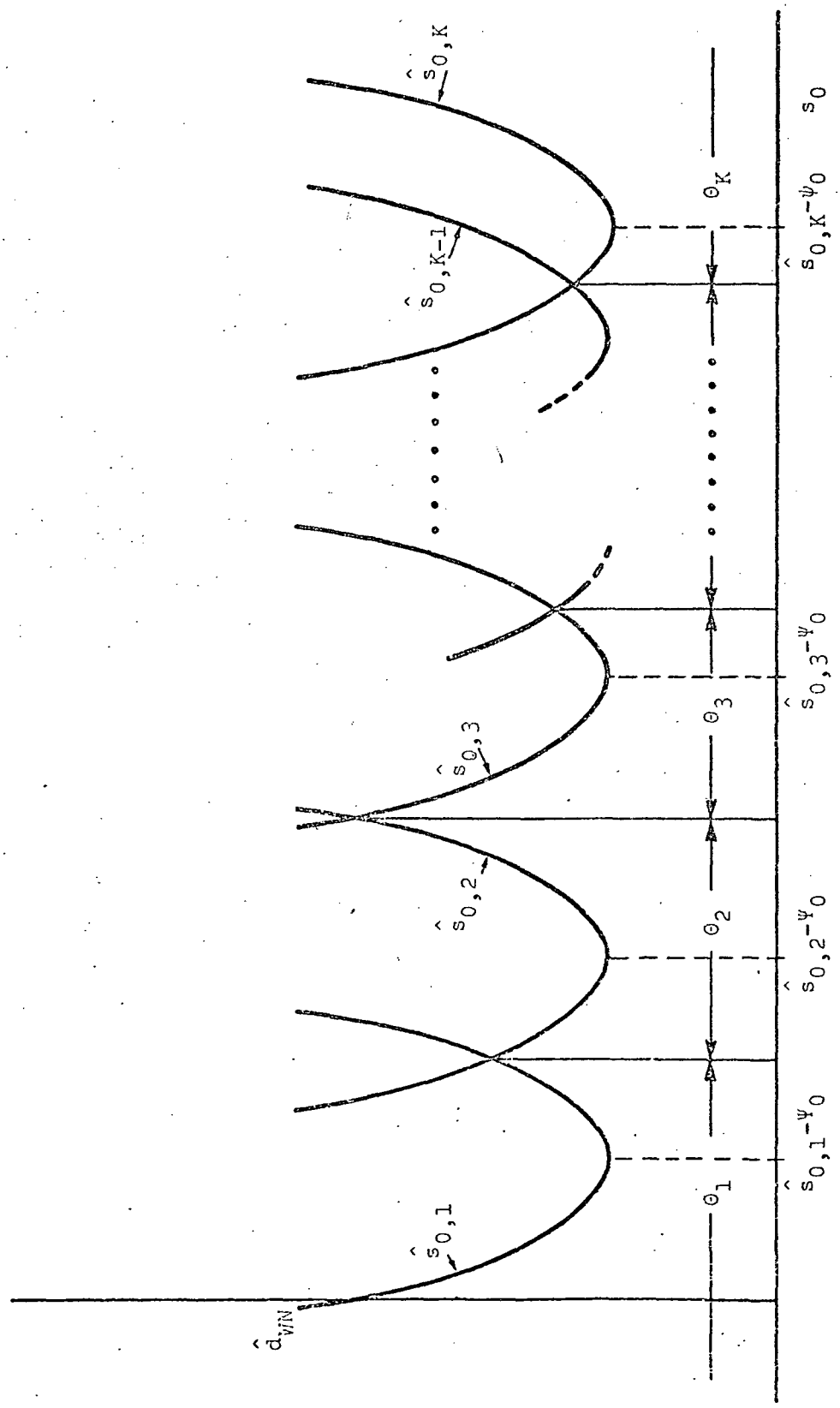



FIGURE 3.1

BLOCK-1 ENCODING REGIONS

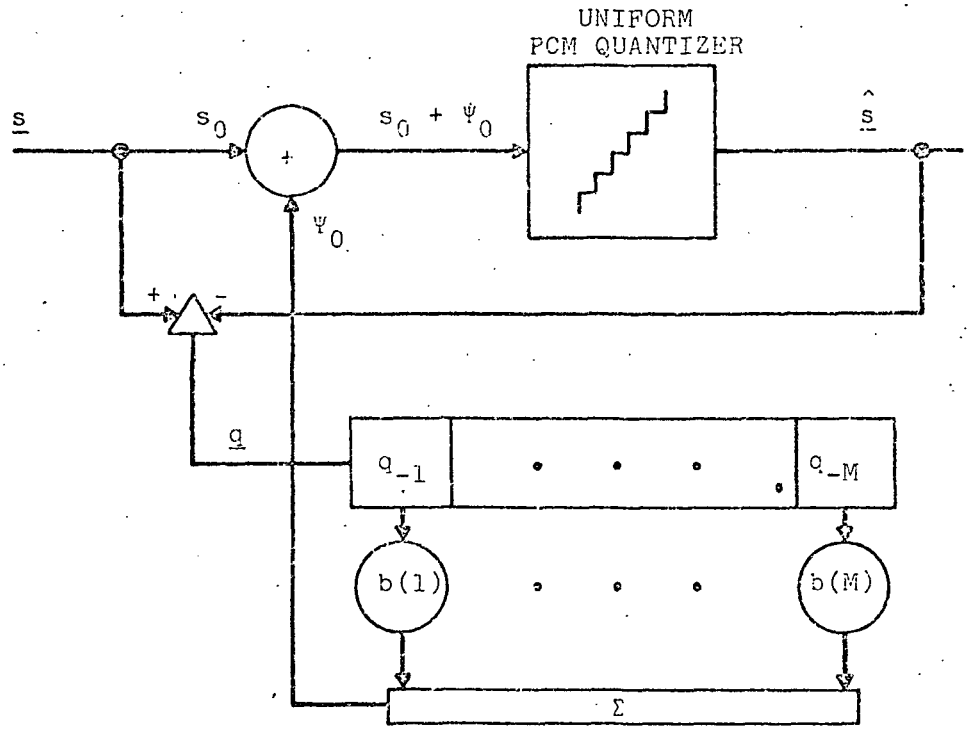



FIGURE 3.2
BLOCK-1 PCM

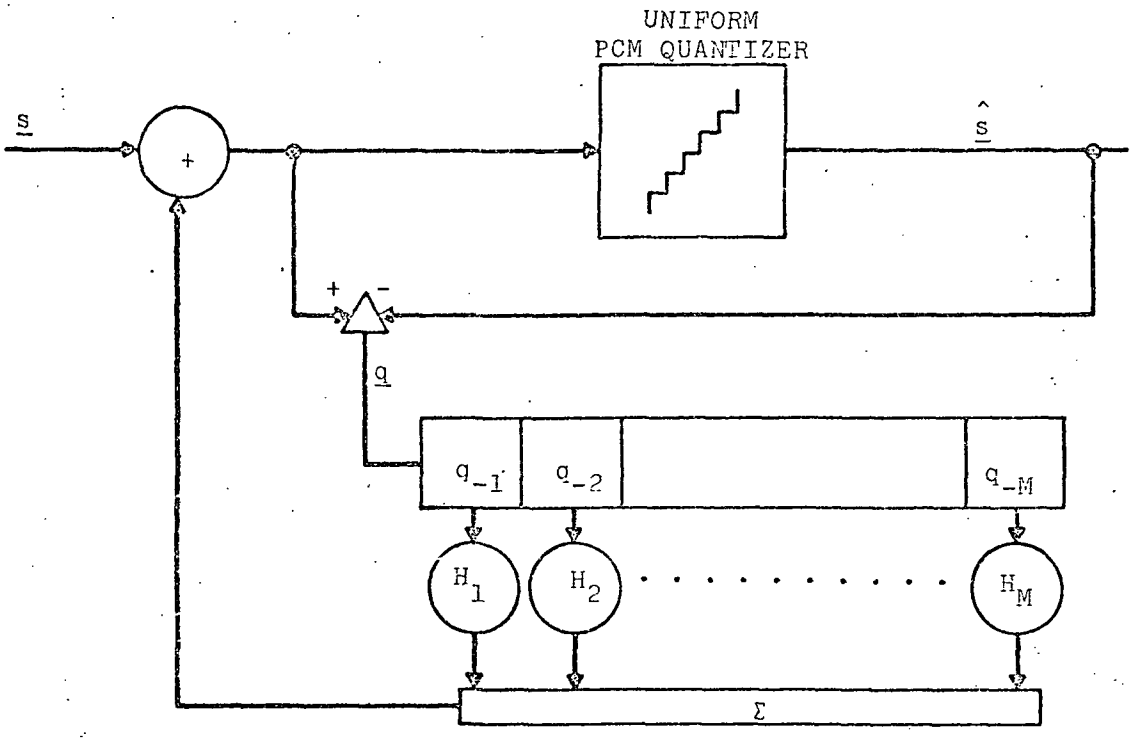



FIGURE 3.3
NOISE FEEDBACK PCM

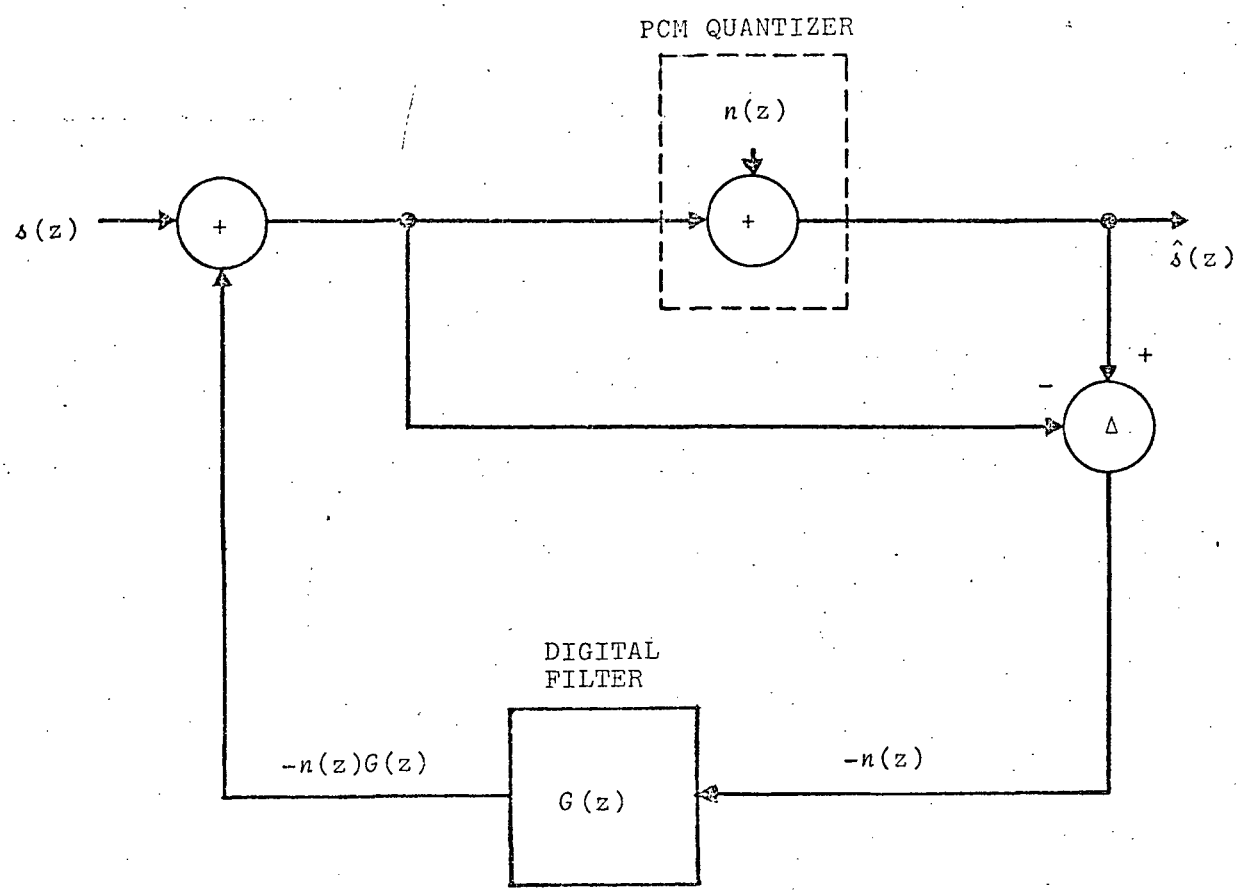



FIGURE 3.4a LINEARIZED MODEL OF
NOISE FEEDBACK PCM

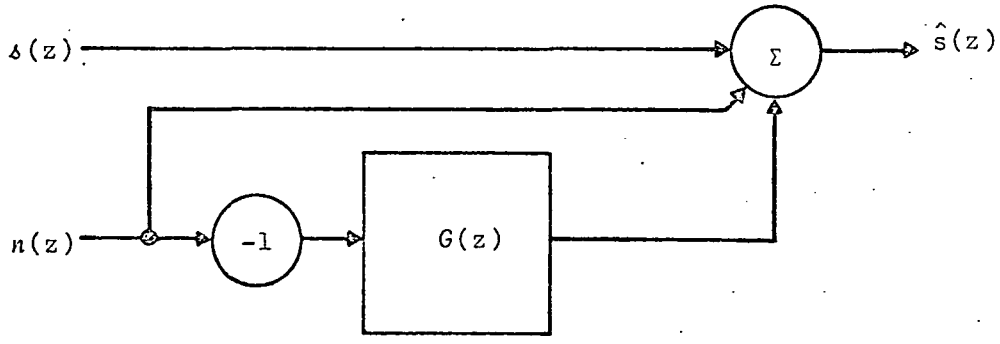



FIGURE 3.4b
EQUIVALENT CONFIGURATION OF LINEARIZED NOISE FEEDBACK PCM

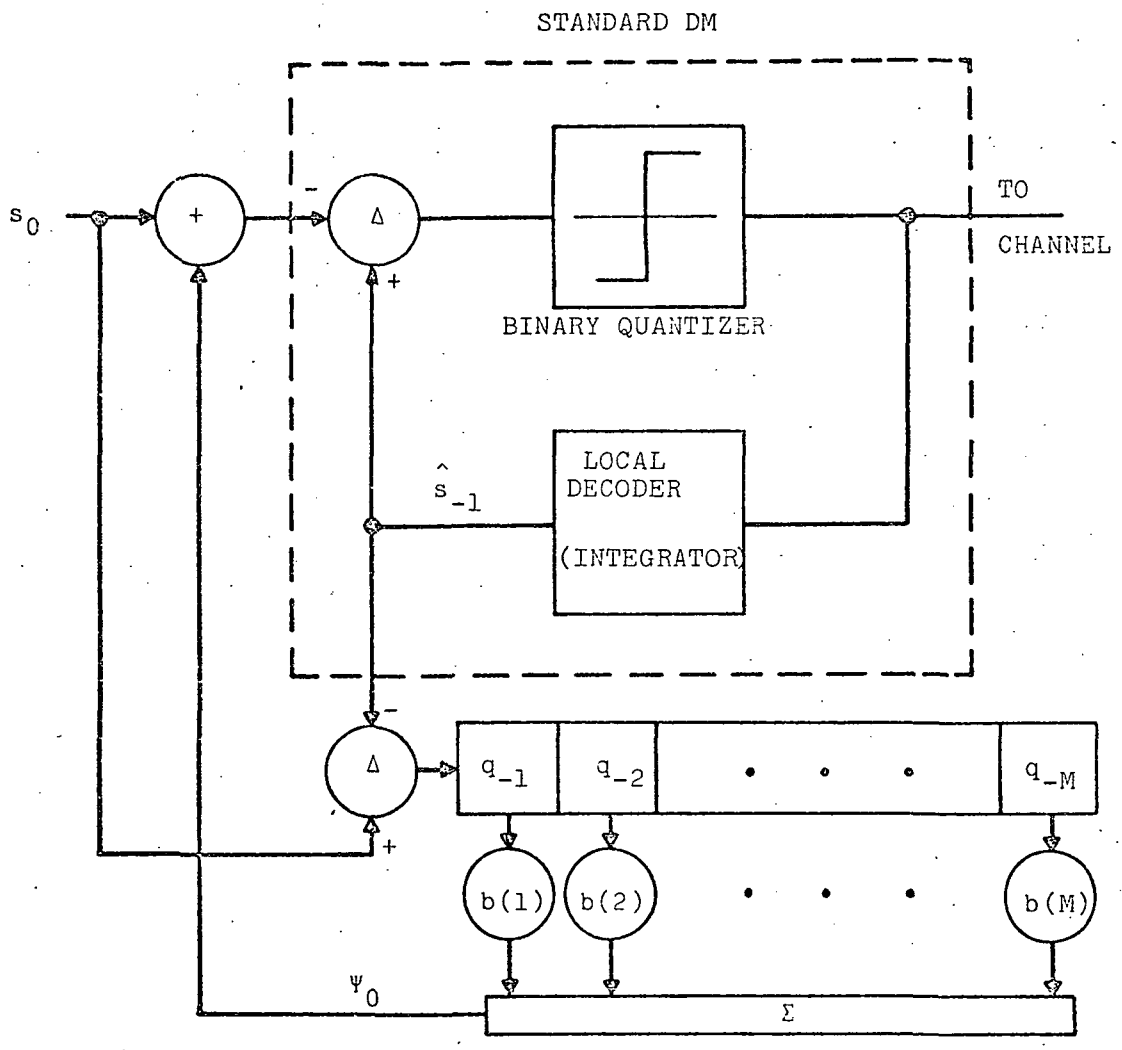



FIGURE 3.5
BLOCK-1 DM

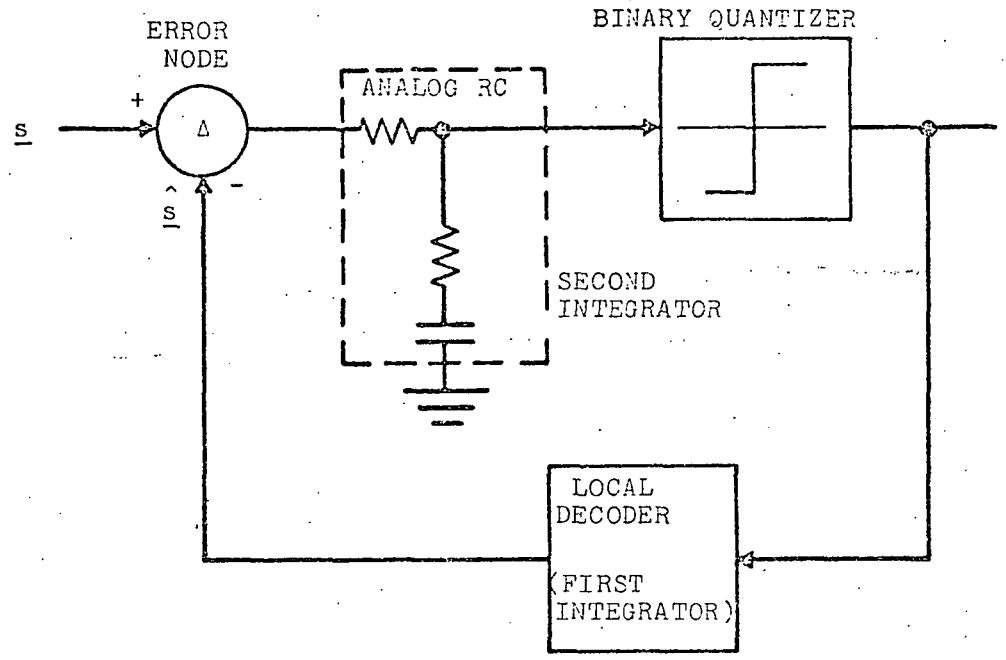



FIGURE 3.6a

DM WITH A SECOND INTEGRATOR

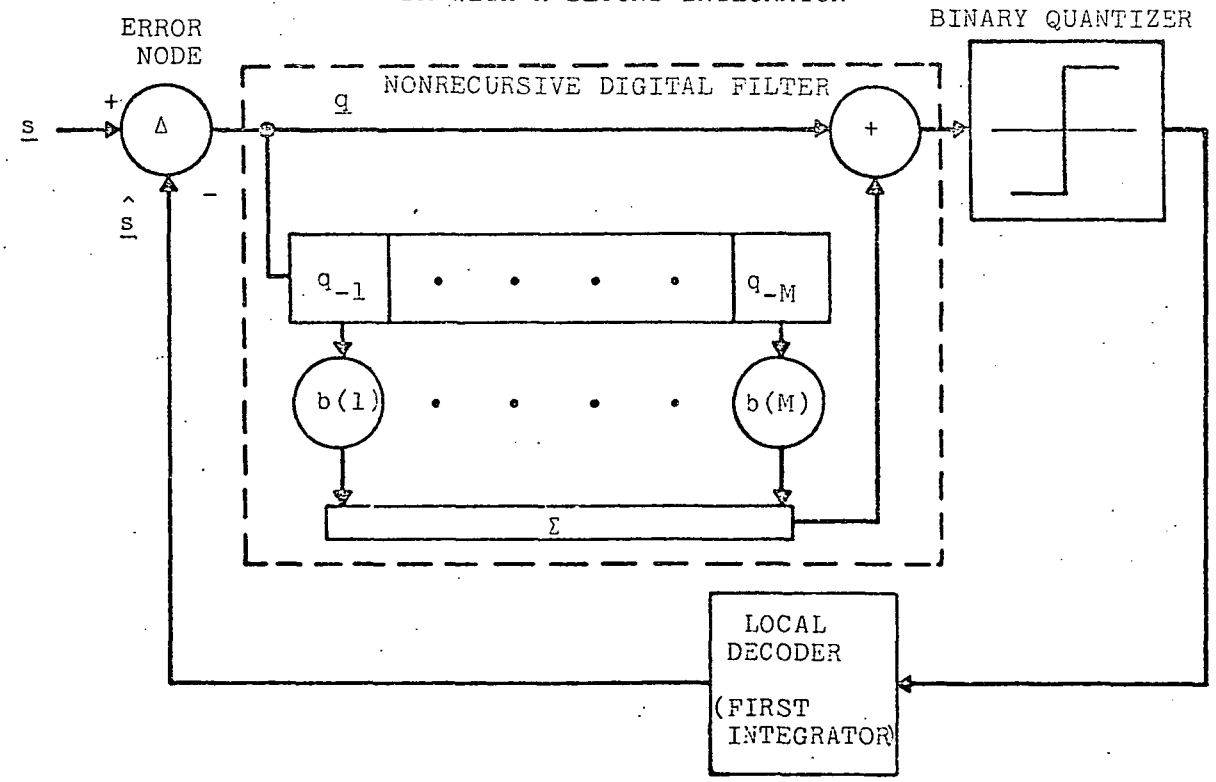



FIGURE 3.6b

EQUIVALENT FORM OF BLOCK-1 DM

# Chapter IV

## 4.1 General Considerations of Block Encoding

When the block length is greater than one, sufficient complexity is introduced into the encoding to make the implementation of an algorithm a significant problem of itself. Contrary to block-1, the block-2 and higher dimensional decision rules cannot be reduced to an additive predistortion of the input sample in an otherwise standard encoder. The reason is that when two or more input samples are jointly encoded, their interrelationship, which is now important, causes the $\Theta_\nu$ volumes to have complicated shapes, and this does not allow a simple threshold decision for the individual channel letters in the output block. Instead, the output letters are given by logical functions of the boundary test (2.3.13) for the $\binom{K^N}{2}$ pairs of candidate reconstruction sequences.

Besides the central question of how to implement the logic to make these decisions, there are ancillary problems of timing, buffering, and generating the bias vector, $\beta\vec{Q}_p$, which must be taken into account in a practical encoder design. Also, it is very desirable to be economical of computation, and to keep to a minimum the generation of redundant data in the encoding process. These points are discussed with regard to the design of a general block encoder in this chapter.

Before taking up machine processing, however, consider how one could do the encoding by hand. In the most pedestrian manner, $\hat{d}_{WN}$ is computed for all $K^N$ outcomes and $\underline{c}_0^{N-1}$ which gives the minimizing $\hat{\vec{S}}_f$ is selected. Alternatively, a definite savings in effort is gained by an ordering procedure whereby the nearest of an arbitrary initial pair $\hat{\vec{s}}_{f,1}$, $\hat{\vec{s}}_{f,2}$ is determined through (2.3.15), and it is then compared with the third candidate. Then the fourth is compared with the best of this pair, and so on. At each step, the better reconstruction sequence of the two is retained for testing with the next one on the list, and the other is discarded since it could not be the optimum choice. This method takes $K^N-1$ computations, the last of which yields the survivor, and therefore the best $\hat{\vec{S}}_f$.

Although nearly the same number of calculations is required, the total work involved with the latter method usually is much less. When the receiver is stationary, or even in some adaptive cases, $\vec{\Delta}_{\nu\xi}$ and $\vec{\Sigma}_{\nu\xi}$ are effectively constant vectors. Furthermore, several pairs might have a common $\vec{\Sigma}$, and others a common $\vec{\Delta}$. A catalogue of $\vec{\Delta}^T$, $\vec{\Delta}^T B_N \vec{\Sigma}$, and $\vec{\Delta}^T B_N$ can be prepared for all the pairs, thus avoiding much of the redundancy inherent in evaluating (2.3.15) repeatedly for each test.

The entire encoding procedure is easily programmed on a digital computer. When $K^N$ is large, however, storage requirements become significant. For greatest efficiency, a separate table is required which stores, for each pair, the

above three parameters. Then each test only involves forming the two inner products $\vec{\Delta}^T \beta \vec{Q}_p$ and $\vec{\Delta}^T B_N \hat{\vec{S}}_f$, which are summed to compare with $\vec{\Delta}^T B_N \vec{\Sigma}$. Totalling the memory used, there are three tables,* each storing $\binom{K^N}{2}$ N element arrays. This amounts to $\frac{3}{2} N(K^{2N}-K^N)$ memory locations, which is not a small number in many interesting cases.

A practical hardware realization of block encoding precludes such a vast amount of storage, and so some efficiency must be sacrificed in order to do machine processing without a programmed computer. Also, if the channel letters are generated in a block, buffering must be provided to present them at a uniform rate to the channel.

The encoder design given here is an attempt to satifsy these requirements, and also strike a reasonable balance between the amount of storage, computational effort, and complexity of the control logic. It makes the individual output digit decisions within a block in a sequential fashion, whereby the updated system state affects the next digit decision. In this sense, it is reminiscent of the PCM quantizer which maps input amplitude into a block of binary digits by an iterative process which uses a single polarity detector. After a bit is encoded by the detector, the "value" of the bit is subtracted from the sample before the next encoding, etc.

---

* A fourth table, containing the channel letter sequences, can be eliminated if $\underline{c}(\nu)$ can be constructed from the index $\nu$.

## 4.2 Encoding Regions for Block-2 DM, a Comparative Analysis

While it is quite difficult to visualize the decision regions when the encoding space is three dimensional, and hopeless when $N > 3$, to do so for block-2 at least is invaluable for an intuitive comprehension of the decision rule. Block-2 DM is chosen for an illustrative example because it is easily described, and is one of the systems which was simulated in the experimental phase of this work.

The source sample space for block-2 is, of course, the $s_0$, $s_1$ plane. Since $K = N = 2$, there are four candidate vectors $\hat{\vec{S}}_f$, and corresponding to each is an optimum source vector given by

$$\vec{S}_f\text{opt} = \hat{\vec{S}}_{f,\nu} - \vec{\Psi} \quad , \quad \nu \Longleftrightarrow \underline{c}_0^1 = (1,1),\ (1,-1),\ (-1,1),\ (-1,-1) \tag{4.2.1}$$

These are enumerated in Table 4.1, assuming a constant step size $\delta$, and plotted on the encoding plane in Figure 4.1.

To graph the cost, which is now a joint function of $(s_0,s_1)$, requires a third axis. As represented in the perspective of Figure 4.2, the cost surface is an elliptic paraboloid centered about $\vec{S}_f\text{opt}$ with minimum value, at the vertex, equal to $\lambda'$ (c.f. 2.3.5). The intersection of the cost surface with a plane

$$\hat{d}_{WN} = \Lambda > \lambda'$$

is an ellipse with semiaxis inclined 45° to the $s_0$, $s_1$ coordinates. The intersection with planes $s_o \pm s_1 = \text{const.}$ are parabolas. These facts are immediate consequences of the $B_N$ coefficients.

Now imagine four such separate congruent cost functions, each located about one of the points $\vec{S}_f$opt. This is described in Figure 4.3 by projecting to the encoding plane the families of constant cost ellipses generated by passing several parallel planes at different heights $\Lambda$ through the four sheet complex. It is now a straightforward matter to ascertain the encoding region boundaries, in a geometric procedure, using the property that the ellipse semiaxis, ie, its size, is an increasing function of $\Lambda$.

A given point is in the half space $\Theta_{\nu<\xi}$ if and only if the $\nu$ ellipse which passes through that point is smaller than the $\xi$ ellipse which passes through it. Here some obvious abbreviations in notation were adopted. The $\nu \lessgtr \xi$ border is therefore the locus of intersections of equal size $\nu$ and $\xi$ ellipses. It has already been shown this locus must be a straight line, which is now clearly seen to pass through the midpoint between $\vec{S}_\nu$opt and $\vec{S}_\xi$opt, being tangent to the two ellipses sharing the midpoint. Finally, an individual encoding region is just the intersection of three half planes, $\Theta_{\nu<\xi}$, $\xi \neq \nu$. The complete picture is given in Figure 4.4, where the encoding boundaries have been

added. It should be emphasized that these encoding regions, whose shapes are a function only of $\vec{\Delta}_{\nu\xi}$ and b(1),* are fixed to within a translation. This means that the usually complicated logic which determines the region in which $\vec{S}_f$ lies can be time invariant.

An interesting fact revealed by this view is that the boundary between regions corresponding to (-1,-1) and (1,1), which is not drawn, never enters into the encoding. This is not quite an oddity, as generally only the borders separating nearest neighbors are important. However, such details are not apparent in a purely formal analysis, and a search for don't care boundaries using only the defining equations is far from trivial.

Further insight is gained by contrasting block-2 DM decision regions with the corresponding partitioning of the sample space caused by a standard encoder. This is done by considering two successive standard DM encodings as comprising a block. The first channel digit is decided by the sign of $s_0-\hat{s}_{-1}$, and so all $(s_0,s_1)$ points in the right half plane $s_0 > \hat{s}_{-1}$ in Figure 4.6 encode to +1 for the first digit. In the right half region, the second digit is given by the sign of $s_1 - (\hat{s}_{-1}+\delta)$, and in the left by the sign of $s_1 - (\hat{s}_{-1}-\delta)$. The boundaries are drawn accordingly, and

---

* When N = 2. For N = 3, b(1) and b(2), as well as $\vec{\Delta}_{\nu\xi}$, fix the configuration, etc. These boundaries are drawn for unity in-band weighting with a sampling rate ratio of 8.

the four points are the optimum $\vec{S}_f$ values in their respective regions.

Comparing block-2 DM with these standard encoding regions makes evident the two salient properties of the frequency weighted noise rule. First, the frequency weighting of the noise shifts the origin of the sample space (relative to the boundaries) according to the past errors. This is the N-dimensional biasing which in part causes the error sequence to assume the desired autocorrelation structure, whereby the noise spectral shaping is a direct result. Second, the interactive effect of a joint (block) decision on the encoding rule is made evident. The standard encoding regions have rectilinear boundaries, indicating the relative insensitivity of the $\hat{s}_1$ decision to the value of $s_0$. Of course, $s_1$ never affects the prior choice $\hat{\vec{s}}_0$, and $s_0$ only grossly interacts with $\hat{s}_1$, as reflected in the offset of the $\hat{s}_1$ boundary from left to right. On the other hand, the block-2 boundaries display the now important interaction between the values of $s_0$ and $s_1$, and how the reconstruction is chosen to give a desirable noise sequence, taking into account both input values.

A more comprehensive view is seen by subjecting block-1 DM to the same analysis, ie, examining the two dimensional decison regions formed as the result of a double block-1 DM iteration. The quad of $\vec{S}_f$opt points looks like that in Figure 4.1, however the optimum error vector, which shall be denoted $-\vec{\Psi}$(block-1) to indicate that it is still

block-1 derived, will be different. Certainly $\Psi_o$(block-1) is still as given in (3.1.2); there is no predictive interaction from the $s_1$ sample back to $c_0$. Also,

$$\Psi_1(\text{block-1}) = \frac{b(1)}{b(0)}q_0 + \ldots + \frac{b(M)}{b(0)}q_{1-M} \qquad (4.22)$$

is just the "updated" version of its predecessor $\Psi_o$(block-1). However, for an optimum point it must be that

$$s_0 - \hat{s}_0 = q_0 = -\Psi_0(\text{block-1}) = -\sum_{j=1}^{M}\frac{b(j)}{b(0)}q_{-j} \qquad (4.2.3)$$

so that substituting in the above gives, for the optimum in $s_1$,

$$s_1 - \hat{s}_1 = q_1 = -\Psi_1(\text{block-1}) = -\sum_{j=1}^{M}\frac{b'(j)}{b(0)}q_{-j}$$

where

$$\left.\begin{aligned} b'(j) &= b(j+1) - b(j)b(1) \quad , \quad 1 \le j \le M-1 \\ \text{and} \\ b'(M) &= -b(M)b(1) \end{aligned}\right\} \qquad (4.2.5)$$

Now compare with block-2. The corresponding vector $B_2^{-1}\beta\vec{Q}_p$ has error term coefficients

$$b''(j) = \frac{b(j+1) - b(j)b(1)}{1 - b^2(1)} \qquad (4.2.6)$$

for $\Psi_1$, and

$$b''(j) = \frac{b(j) - b(j+1)b(1)}{1 - b^2(1)} \qquad (4.2.7)$$

for $\Psi_o$. There is no strong correspondence between the first digit parameters, but it is clear that

$$\Psi_1(\text{block-1}) = \Psi_1[1 - b^2(1)], \qquad (4.2.8)$$

which suggests that for the second digit, at least, block-1 repeated is roughly akin to block-2.

This is made strikingly apparent in the form of the encoding boundaries, Figure 4.6. The $c_o = +1$ region is the right half plane

$$s_o + \Psi_o(\text{block-1}) > \hat{s}_{-1} \qquad (4.2.9)$$

but the second digit borders are given by the updated equation

$$s_1 + \Psi_1(\text{block-1}) = \hat{s}_o = \hat{s}_{-1} \pm \delta \qquad (4.2.10)$$

which can be rewritten, using the definition for $q_o$, as

$$b(0)s_1 + b(1)s_o = \hat{s}_{-1} \pm \delta + \text{const.} \qquad (4.2.11)$$

The constant depends on the past errors, but this form reveals that the boundary is a straight line with slope $-\frac{b(1)}{b(0)}$. It is constructed by noting that it must pass through the midpoint of the vertical line connecting the pair of optimum $s_1$ points, in each $c_o$ region, because of the minimum distance property of block-1 encoding.

It will be shown later that the analogous block-2 boundaries, ie, (1,1) vs. (1,-1) and (-1,1) vs. (-1,-1) have the same slope -b(1), and as Figure 4.4 indicates they also

pass through the midpoints of the vertical lines connecting the optimum loci. Thus the second digit boundaries of block-1 repeated are in remarkably close agreement with those of true block-2 encoding.

The differences, however, occur close to the center of the optimum $\vec{S}_f$ region, loosely defined as about or within the parallelogram with vertices $\vec{S}_f$opt. That is, when the source sample vector departs greatly from the low encoding cost region, block-1 and block-2 decisions are more or less equivalent. On the other hand, block-2 exhibits the look-ahead property inherent to itself and higher dimensional encodings whereby large errors in future digits can be foreseen and diminished by virture of the joint decision over several preceeding digits.

It is interesting to review, by way of the encoding region diagrams, the progression from standard DM, through block-1, and finally to block-2. The big improvement, of course, is from standard to block-1, where past errors now enter into the decision in the form of a translation of boundaries. The tilted second digit boundaries in repeated block-1 point up the sensitivity of $c_1$ to the value of $s_0$, and not just $c_0$. A significant refinement, however, is seen in the transition from block-1 to block-2, where both look ahead and small error optimization are qualities definitely discernible in the encoding region structure.

Of course, the end result in a practical sense is an almost infinite number of iterations of the block-N encoder, processing an entire message. One is interested, therefore, in how this local optimization over L samples envolves into a global effect. The previous example gives a slight hint, but no hope of analysis was seen. Indeed, only recently have certain inroads been made in analyzing the simplest form of standard DM, and this for a very restrictive class of (unrealistic) inputs. It is hoped, then, that an empirically based analysis is sufficient to justify, if not prove, the claim that the properties attributed to the local decision, ie, minimizing weighted error, do propagate into a similar overall reduction of the same quantity.

## 4.3 Stream Mode Block Encoder

A block encoder which processes digits in a stream fashion, both accepting source samples and producing channel letters in a continuous flow, will now be described. Input and output buffering is thereby intrinsic to the operation of this device. Of course, there is still an N letter delay; $c_j$ is generated after $s_{j+N}$ is read in. It will be possible to easily select any block length encoding, up to the maximum designed for, with no alteration of the decision making logic. Also, nearly all of the available redundancy has been utilized in the mechanism for computing the $\beta\vec{Q}_p$ vector, in order to enhance efficiency.

The basic principle of operation is to encode symbol by symbol, in a sequential manner, starting with the first unknown output $c_0$ and progressing to the end of the block. The candidate outputs $\underline{c}_0^{N-1}$, are partitioned into $K^{N-1}$ groups according to the value of $c_0$, and that group containing the optimum output is sought. In making this search, notice that no tests need to be made between candidate $\hat{\vec{S}}_f$ sequences belonging to the same $c_0$ class.

Once the first digit of the output block is determined, so is the respective reconstruction value $\hat{s}_0$, and hence the error $q_0$. Now the second output digit is to be decided, and this is still to be done by minimizing the quadratic form which gives $\hat{d}_{WN}$. However, this time there

are M+1 fixed vector components $\{q_{-M},\ldots,q_0\}$, and N-1 "free" ones $\{q_1,\ldots,q_{N-1}\}$, which can now assume only $K^{N-1}$ distinct configurations. Encoding the second digit, then, is equivalent to a first digit encoding wherein M+1 past errors are used and the block length is N-1. Again, the channel outputs are partitioned into $K^{N-2}$ equivalence classes according to the value of $c_1$, and the class containing that output which minimizes $\hat{d}_{WN}$ is found. $\hat{s}_1$ is thereby determined, giving rise to $q_1$, and this places one more undetermined error into the known category.

The process, then, repeats until at the start of the last step all components of $\hat{\underline{S}}_f$ have been determined, except $\hat{s}_{M+N-1}$. At each iteration, the dimensionality of the source sample space decreases by one, and the memory length of past errors increases by one. However, at each step in the process the very same distance measure in the original N+M dimensional space is minimized to choose that channel digit, so the result of this iterative procedure is the identical $\hat{\underline{S}}_f$ or $\underline{c}_0^{N-1}$ which would have been found by any other technique that minimizes $\hat{d}_{WN}$.

To rephrase this somewhat, the initial step is to make a first digit only decision in a block-N encoding, where there are M past errors. The result of this decision is used to update $\hat{\underline{S}}$ one time slot, which produces a new error $q_0$. Then, using the M+1 past errors, a first digit

decision is made for a block-(N-1) encoding, generating $\hat{s}_1$ and $q_1$. This continues until the last encoding, which is just a block-1 decision using M+N-1 past errors. This completes the current block encoding, and the next step would be a block-N decision on the first digit of the next block, etc.

A simple example will serve to relate this algorithm to the actual device. Consider the functional block diagram in Figure 4.7, which is a stream mode block-2 DM encoder. In place of the "block box" which would be used to make a block-2 joint decision, there are two single digit decision devices, with a switch to select the output of either one. The upper, which is initially connected to the channel, will determine the first bit only in a block-2 decision. By comparison with Figure 4.3, this is suggested by the peculiarly shaped line segment boundary drawn in the box, which signifies distinguishing the $c_0 = +1$ from the $c_0 = -1$ first digit regions.

A two stage analog shift register is provided to store the input sample pair, and the connections have been drawn between the register cells and the first digit block-2 device.

In conjunction with their respective input samples, the components of $\beta\vec{Q}_p$ have been indicated as inputs to the first digit encoders, as well as inputs from the local decoder output $\hat{s}_{-1}$. They might otherwise have been included as

individually adding to each input before the decision device. The circuitry for generating $\beta\vec{Q}_p$ has been omitted here for clarity, but it will be discussed in the next section.

The lower decision box is a block-1 encoder, and it makes the last (second) decision of the block. After the first digit is decided, the local decoder updates the reconstruction to $\hat{s}_0$, and the subtractor feeds $q_0$ to the past error bias network, which computes an updated $\beta\vec{Q}_p$ vector. The output switch now moves to connect the block-1 decision device to the channel. The source sample shift register is cycled (signals move to the right) and now contains $s_2$ and $s_1$, where $s_1$ is now the input to the block-1 box.

This is the desired configuration; only $s_1$ is left to encode since $\hat{s}_0$ has already been established. This may seem wrong at first, for $s_0$ should also enter into the encoding of the second digit. But it has in a subtle way. The block-1 decision uses an updated $\beta\vec{Q}_p$ component which is a function of $q_0$, and $q_0$ is given by $s_0$. Thus the vertical line in the block-1 box, which symbolizes the threshold decision on $s_1$, is biased left or right as a function of $s_0$, through the intermediate variable $q_0$.

After the block-1 encoding is performed, generating $s_1$ and $q_1$, the $\beta\vec{Q}_p$ vector is again updated and the input register is cycled once more to now contain $s_3$ and $s_2$. The switch commutes back to the block-2 device output, and the

encoder is automatically ready for a new block encoding. Note that no block interval timing is necessary, and all processing, updating of shift registers, etc., takes place at the basic channel symbol rate.

The extension of this construction to larger block lengths, as visualized in Figure 4.8, is simple. There would be N devices; a block-N first digit encoder, block-(N-1) first digit encoder, ..., block-1 encoder whose outputs terminate on sequential stators of a commutating switch. The wiper, which is connected to the channel, visits them in the order: Block-N, ..., Block-1, Block-N, ..., Block-1, ... because of the circular nature of the arrangement. An N cell input register is provided, with N connections to the block-N box, N-1 connections to the block-(N-1) box, etc. Also, each decision box gets the corresponding number of $\beta\vec{Q}_p$ components as inputs, as well as the local decoder output, although when N is large it might be advantageous to just pre-add $\beta\vec{Q}_p$ to the input and then store the result in a separate register.

$\beta\vec{Q}_p$ cannot be simply bussed to the first digit encoders as are the source samples, because each device requires a slightly different set of bias values. For example, the third component of $\beta\vec{Q}_p$ which feeds the block-5 device is not equal to the third $\beta\vec{Q}_p$ component for the block-4, or block-3 device, etc.

The reason is that each device makes a decision which uses a different number of past errors, ie, M+N-N′, for the block-N′ unit. On the other hand, the differences among the various $\beta\vec{Q}_p$ components of like index occur in the inclusion of more or less lowest order terms of the summation, ie, the terms involving the errors farthest in the past. The $\beta\vec{Q}_p$ vector could be bussed to the first digit encoders if only the vector for block-N were used, and this approximation may well be justified by the simplicity which is gained. Performance would be expected to degrade, although perhaps very slightly, since this is going in the direction of taking fewer average past errors into consideration for each encoded digit.

A property of this stream mode encoder which makes it particularly useful for experimentation is block length selection. Any block length $N' \leq N$ can be established merely by causing the switch to bypass the first digit encoders whose order is higher than N′. No other alternations are necessary. Of course, the effective memory length for past errors will be M+N-N′, assuming the exact $\beta\vec{Q}_p$ components are used. Thus the trade-off between encoding dimensionality and the past error memory, keeping total storage M+N fixed can be explored easily. This is one of many interesting questions for further research.

## 4.4 Computing the Past Error Bias Vector

Every N-dimensional decision, whether it is an entire block encoding or just one digit in a block, requires the past error bias vector $\beta\vec{Q}_p$. This N-component vector must be recomputed for each successive encoding, after the error sequence is updated. Because of the special structure of $\beta$, wherein its rows are, roughly speaking, translates of each other, to compute each element of $\beta\vec{Q}_p$ in a separate summation (or separate transversal filter) entails much redundant work. However, by suitably using the symmetry of $\beta$, the computation can be done with work proportional to M, rather than NM. When M is large, typically 20 or 30, there is a distinct advantage with the method which follows over generating the data simultaneously in a parallel arrangement of filters.

To implement the stream mode encoder without any approximations, it is necessary to provide a distinct, updated bias vector for each of the first digit encoders. Here again, these are not computed from scratch, but are obtained from summations already performed by adding in the missing lower order terms. It is convenient to introduce a new symbol, $\vec{\chi}$ to represent $\beta\vec{Q}_p$, and use a prefix subscript to denote the particular digit in the block to which the bias vector applies. For example, ${}_2\chi_1$ is the second to the last element, corresponding to the $s_1$ bias, of the vector pertaining to the third digit encoding, $c_2$.

A simple example will indicate how a single transversal filter operating on the error sequence, such as in a block-1 encoder, can be augmented to yield all the terms in an N-dimensional $\vec{\chi}$ vector. Consider block-2, where the bias* for the first digit $c_0$ is

$$ {}_0\vec{\chi} = \begin{bmatrix} {}_0\chi_1 \\ {}_0\chi_0 \end{bmatrix} \quad \begin{bmatrix} \sum_{j=1}^{M} b(j+1)q_{-j} \\ \sum_{j=1}^{M} b(j)q_{-j} \end{bmatrix} \qquad (4.4.1) $$

and the bias for the $c_1$ encoding is

$$ {}_1\chi_0 = \sum_{j=1}^{M+1} b(j)q_{1-j} \qquad (4.4.2) $$

Figure 4.9 shows how the three pieces of data might be derived, using a single transversal filter with tap gains b(2),...,b(M), as the basic component. Square blocks, including those of the filter shift register, represent memory cells wherein analog samples are stored one cycle. The data enters from the left and exits right, and the current cell contents appear on the lines emanating from the top or bottom of a cell. Thus a single memory cell in series with a lead accomplishes a one cycle delay. Multipliers are drawn as circles, with their value, or gain, as inscribed.

Memory cell contents have been labelled with the signals stored just prior to the $c_0$ encoding. A brief inspection will verify that the outputs labelled ${}_0\chi_0$ and ${}_0\chi_1$

---

* For convenience, the b coefficients have already been normalized with respect to b(0).

do, in fact, equal the sums (4.4.1). Now let the $c_0$ digit be encoded, and cycle the system. The ${}_0\chi_0$ and ${}_0\chi_1$ outputs are now

$$\sum_{j=1}^{M} b(j)q_{1-j} \quad , \quad \sum_{j=1}^{M} b(j+1)q_{1-j}$$

respectively, where it is clear that these may be written down by formally increasing by one the error variable subscripts in (4.4.1). As such, these new sums are not pertinent; they belong to a $c_0$ encoding. However, by adding $b(M+1)q_{-M}$ to the new ${}_0\chi_0$ sum, the ${}_1\chi_0$ output is obtained. This is accomplished, as shown, by a delay of the output of the last multiplier.

The encoder will then choose $c_1$, which results in setting $q_1$. The twice updated memory cells will subsequently contain $q_1,\ldots,q_{2-M}$ in the shift register (including the last one which is not formally part of the M-1 stage filter),

$$\sum_{j=1}^{M} b(j+1)q_{2-j}$$

in the delay cell which feeds ${}_0\chi_0$, and $b(M+1)q_{2-M}$ in the cell feeding ${}_1\chi_0$. Thus the ${}_0\chi_0$ and ${}_0\chi_1$ outputs are the correct bias elements to encode the first digit of the new block, and the next cycle would cause ${}_1\chi_0$ to be the correct bias for the second digit of the second block, etc.

The extension to longer block length is fairly straightforward, although the complexity of the circuit

increases rapidly with N. Much of that increase, however, is caused by requiring a different bias vector for different digits within a block. Next, a block-3 example will be given, and this should be sufficient to indicate the pattern for all the higher block sizes.

Block-3 requires six separate bias terms. Besides the three generated in a block-2 circuit, there are the additional outputs

$$\left.\begin{aligned} {}_0X_2 &= \sum_{j=1}^{M} b(j+2)q_{-j} \\ {}_1X_1 &= \sum_{j=1}^{M+1} b(j+1)q_{1-j} \\ {}_2X_0 &= \sum_{j=1}^{M+2} b(j)q_{2-j} \end{aligned}\right\} \qquad (4.4.2)$$

to be provided. Incidentally, the general expression for these terms is

$${}_uX_v = \sum_{j=1}^{M+u} b(j+v)q_{u-j} \qquad (4.4.3)$$

which is easily determined from the $\beta\vec{Q}_p$ matrix product.

The circuit of Figure 4.10 is offered as one way in which the required Block-3 parameters may be generated in a fairly efficient manner. To check the ${}_0X_0$ output, for example, notice that it is the sum of $b(1)q_{-1}$, and $b(2)$ times the error variable delayed, which would be $q_{-2}$, and the output of the M-2 stage transversal filter twice delayed. Two

cycles in the past, the filter shift register contained (move the variables left twice) $q_{-3},\ldots,q_{-M}$. Hence, the three contributions are

$$\left[b(1)q_{-1} + b(2)q_{-2} + \sum_{j=3}^{M} b(j)q_{-j}\right] \qquad (4.4.4)$$

which certainly equals ${}_0X_0$. The ${}_0X_1$ output is similarly verified by noting that it is the sum of $b(2)q_{-1}$, $b(M+1)q_{-M}$ and the filter output once delayed, giving

$$b(2)q_{-1} + \sum_{j=3}^{M} b(j)q_{1-j} + b(M+1)q_{-M} \qquad (4.4.5)$$

which also checks. The ${}_0X_2$ output is obvious. The rest of the outputs may be examined in a similar fashion.

## 4.5 Stream Mode Digit Encoding Units

It remains to detail the inner works of the first digit decision units of the stream mode block encoder, which until now were considered merely as black boxes. Recall that their function is to determine the value of the first digit only in an N′ dimensional block encoding, where $1 \leq N' \leq N$, and there are N such units. At the extremes, the first unit operates in N dimensions, using N input samples, and the last is merely a block-1 device fed with one source sample. For concreteness, assume k=2, so that one may speak of the output bits as in a binary system. The extension to higher order alphabets will be readily seen.

The given (or observed) source sample vector resides in one of the $2^{N'}$ encoding regions. The object of the unit is to determine the first bit in the output code, actually corresponding to that region. Accordingly, one seeks to dichotomize the regions into two groups based on the first bit of their associated digital codes, and the group containing the region containing the source vector is sought. Whereas an individual encoding region is defined by an intersection of the half spaces of points closer to its $\check{S}_f$opt than all others (cf 2.3.10), each group, G, is a region given by the union of its member regions. Denoting the groups by $G_1$ and $G_{-1}$, and introducing the index sets

$$\left.\begin{aligned} \mathcal{J}_1 &= \{\nu | c_0(\theta_\nu) = 1\} \\ \mathcal{J}_{-1} &= \{\nu | c_0(\theta_\nu) = -1\} \end{aligned}\right\} \qquad (4.5.1)$$

one can see that

$$G_1 = \bigcup_{\nu \in \mathcal{J}_1} \theta_\nu = \bigcup_{\nu \in \mathcal{J}_1} \bigcap_{\xi \in \mathcal{J}_{-1}} \{\vec{S}_f | d(\vec{S}_f + \vec{\Psi}, \hat{\vec{S}}_{f,\nu}) < d(\vec{S}_f + \vec{\Psi}, \hat{\vec{S}}_{f,\xi})\} \tag{4.5.2}$$

which indicates that boundaries need only be considered between pairs of reconstruction points in opposite groups.*

Equation (4.5.2) suggest the computation procedure which is used. The truth of each inequality, viewed now as a logical statement with the given $\vec{S}_f$, $\vec{\Psi}$, and appropriate pair $\hat{\vec{S}}_{f,\nu}$ and $\hat{\vec{S}}_{f,\xi}$ inserted, is determined in separate sub units, each employing a simple threshold detector. The Boolean outputs of these sub units are then combined in a logic circuit which performs the and/or operations corresponding to the union and intersections of (4.5.2).

The number of sub unit threshold detector circuits, and the size of the logic combiner, grows rapidly with $N'$. Considering only binary, still, there are $2^{N'-1}$ regions in each group, which makes for $(2^{N'-1})^2 = 2^{2N'-2}$ sub units.

The total number of sub units in the entire stream mode encoder is therefore

$$\sum_{N'=1}^{N} 2^{2N'-2} = \frac{4^N - 1}{3}. \tag{4.5.3}$$

*Of course, the same region is obtained by taking the union over $\theta_{\nu \in \mathcal{J}_1}$ where $\theta_\nu$ is given by 2.3.10, but this includes unnecessary tests between pairs of points belonging to the same group.

However, this compares favorably with the figure

$$\binom{2^N}{2} = \frac{4^N - 2^N}{2} \tag{4.5.4}$$

which is 50 percent greater in the limit of large N. The latter is the number of boundaries (pairs of candidate reconstruction pairs) which have to be evaluated to do the entire block-N encoding in one fell swoop by a super logic circuit which ascertains that $\Theta_\nu$ for which all inequalities in eq. 2.3.10 are true.

The reason for fewer tests, or boundaries to be checked, with the stream mode encoder is that irrelevant boundaries are automatically found and ignored as more and more of the output digits are determined.

The $N' = 2$ unit in block-2 DM will now be designed as an illustrative example. First, the four boundary equations are obtained, and then combined appropriately in a logic circuit. This exercise also illustrates how to work with the boundary equation in an efficient way.

Begin with the test between output codes 1, 1 and -1, 1 which correspond respectively to reconstructions (cf table 4.1)

$$\hat{\underline{s}}_{f,\nu} = \begin{bmatrix} \hat{s}_{-1} + 2\delta \\ \hat{s}_{-1} + \delta \end{bmatrix}, \quad \hat{\underline{s}}_{f,\xi} = \begin{bmatrix} \hat{s}_{-1} \\ \hat{s}_{1} - \delta \end{bmatrix} \tag{4.5.5}$$

Employing the definition 2.3.14 one has

$$\vec{\Delta}_{\nu\xi} = 2\delta \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\vec{\Sigma}_{\nu\xi} = \begin{bmatrix} \hat{s}_{-1} + \delta \\ \hat{s}_{-1} \end{bmatrix} \tag{4.5.6}$$

Now the "constants" of the test are evaluated:

$$\vec{\Delta}^T_{\nu\xi}\, B_2\, \vec{\Sigma}_{\nu\xi} = 2\delta\Big[b(0)+b(1)\Big]\Big[2\hat{s}_{-1}+\delta\Big] \tag{4.5.7}$$

$$\vec{\Delta}^T_{\nu\xi}\, \beta\vec{Q}_p = \vec{\Delta}^T_{\nu\xi}\, {}_0\vec{X} = 2\delta\Big[{}_0X_0 + {}_0X_1\Big] \tag{4.5.8}$$

$$\Big(\Delta^T_{\nu\xi}\, B_2\Big)^T = 2\delta\Big[b(0)+b(1)\Big]\begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.5.9}$$

When these are inserted in 2.3.15, the result is*

$$(s_0-\hat{s}_{-1}) + (s_1-\hat{s}_{-1}) + \frac{{}_0X_0+{}_0X_1}{b(0)+b(1)} > \delta \tag{4.5.10}$$

---

* When the b coefficients have been already normalized with respect to b(0) in the ${}_0X_0$, ${}_0X_1$ sums, the denominator should be 1 + b(1)/b(0). See eq. 4.4.1.

which gives the condition for preferring 1, 1 over -1, 1. A similar computation for 1, -1 vs -1, -1 gives the same inequality, except $-\delta$ on the right, which is parallel to the previous boundary because the two pairs share a common $\vec{\Delta}_{\nu\xi}$, as can be seen in Figure 4.4. The remaining two tests are similarly computed to be

$$b(0)\left[s_0-\hat{s}_{-1}\right] + b(1)\left[s_1-\hat{s}_{-1}\right] + {}_0\chi_0 > 0 \qquad (4.5.11)$$

for 1, -1 vs -1, 1 and

$$\left[b(1) + \tfrac{1}{2}\, b(0)\right]\left(s_0-\hat{s}_{-1}\right) + \left[b(0) + \tfrac{1}{2}\, b(1)\right]\left(s_1-\hat{s}_{-1}\right) + {}_0\chi_0 + \tfrac{1}{2}\,{}_0\chi_1 > 0 \qquad (4.5.12)$$

for 1, 1 vs. -1, -1.

Figure 4.11 shows how the foregoing is combined to yield the output bit. The upper drawing represents the subunit for the test 1, 1 vs -1, 1, which is equation 4.5.10, reduced to hardware. The output of the binary quantizer in the subunit is presumed to be a logical value, say 1 or -1. Four such subunits are arranged in parallel in the complete first digit unit, each providing an input to the logic combiner, corresponding to the outcomes of the four boundary tests. The inputs $\vec{s}_f$, ${}_0\vec{\chi}$, and $\delta$ are bussed to each subunit.

Lettering the subunit logic outputs as shown, the combiner output is 1 if A and B are both 1, or if C and D are both 1, and it is -1 otherwise. As noted before, the second test, namely 1, 1 vs -1, -1, is superfluous, and so its subunit may be eliminated. The logic combiner simplifies somewhat to A or (C and D). Inspection of the $c_0$ = +1, -1 boundary in Figure 4.4 quickly verifies this statement.

Breaking down the block encoding to a bit by bit process has the advantage that it lends itself to an approximation whereby computation is reduced by deleting less important boundary tests. Only what are judged to be the significant boundaries are retained, usually those between pairs of points $\vec{S}_f$opt which are close neighbors. For example, the 1, 1 vs -1, -1 test, if it weren't already known to be of no consequence on a geometric basis, would be a likely candidate for deletion in an approximate design.

As a further illustration, consider the encoding space for the first digit of block-3 DM in Figure 4.12. The $c_0$ = +1 points are designated ⊕ and the other half ⊖. Here, the encoding volumes, and their bounding planes, are nearly impossible to visualize. A reasonable approximation, reducing the number of tests from 16 to 13, involves deleting from consideration the distinctions (1,1,1 vs -1,-1,-1), (1,1,1 vs -1,-1,1), and (1,1,-1 vs -1,-1,-1). A cruder, yet perhaps nearly as good, approximation is to consider

boundaries only amongst the central quad (1,-1,1), (-1,1,1), (1,-1,-1), and (-1,1,-1). These choices fall in the realm of engineering judgment, and can only be evaluated by experience.

The effect of an approximating first digit unit is the occasional generation of a "wrong" digit, thereby creating higher weighted noise for that block, at least. However, there are several mitigating factors. First, an altered digit (reversed bit in the binary case) will occur in a close situation where the chosen reconstruction is almost as good as the intended one. Second, the error will tend to be early in the block, where the approximations are introduced, but because of the sequential nature of the encoder, in which the past error influence is constantly updated in order to choose the conditionally best reconstruction, the remaining digits will readjust accordingly to produce the best block output given the altered reconstruction.

In conclusion, it is thought that an educated approximation in the larger N′ digit encoders at worst vitiates the large block size, and probably results in only slight diminution of the benefit of N that large. An algorithm for identifying the important decision planes in a block-N first digit encoding is another problem for further investigation, which is important to a practical realization of these ideas.

| $c_0$, $c_1$ | $\vec{S}_f$opt |
|---|---|
| -1, -1 | $\begin{bmatrix} \hat{S}_{-1}^{-\Psi_1-2\delta} \\ S_{-1}^{-\Psi_0-\delta} \end{bmatrix}$ |
| -1, -1 | $\begin{bmatrix} \hat{S}_{-1}^{-\Psi_1} \\ \hat{S}_{-1}^{-\Psi_0-\delta} \end{bmatrix}$ |
| 1, -1 | $\begin{bmatrix} \hat{S}_{-1}^{-\Psi_1} \\ \hat{S}_{-1}^{-\Psi_0+\delta} \end{bmatrix}$ |
| 1, 1 | $\begin{bmatrix} \hat{S}_{-1}^{-\Psi_1+2\delta} \\ \hat{S}_{-1}^{-\Psi_0+\delta} \end{bmatrix}$ |

TABLE 4.1

$\vec{S}_f$opt FOR BLOCK-2 DM

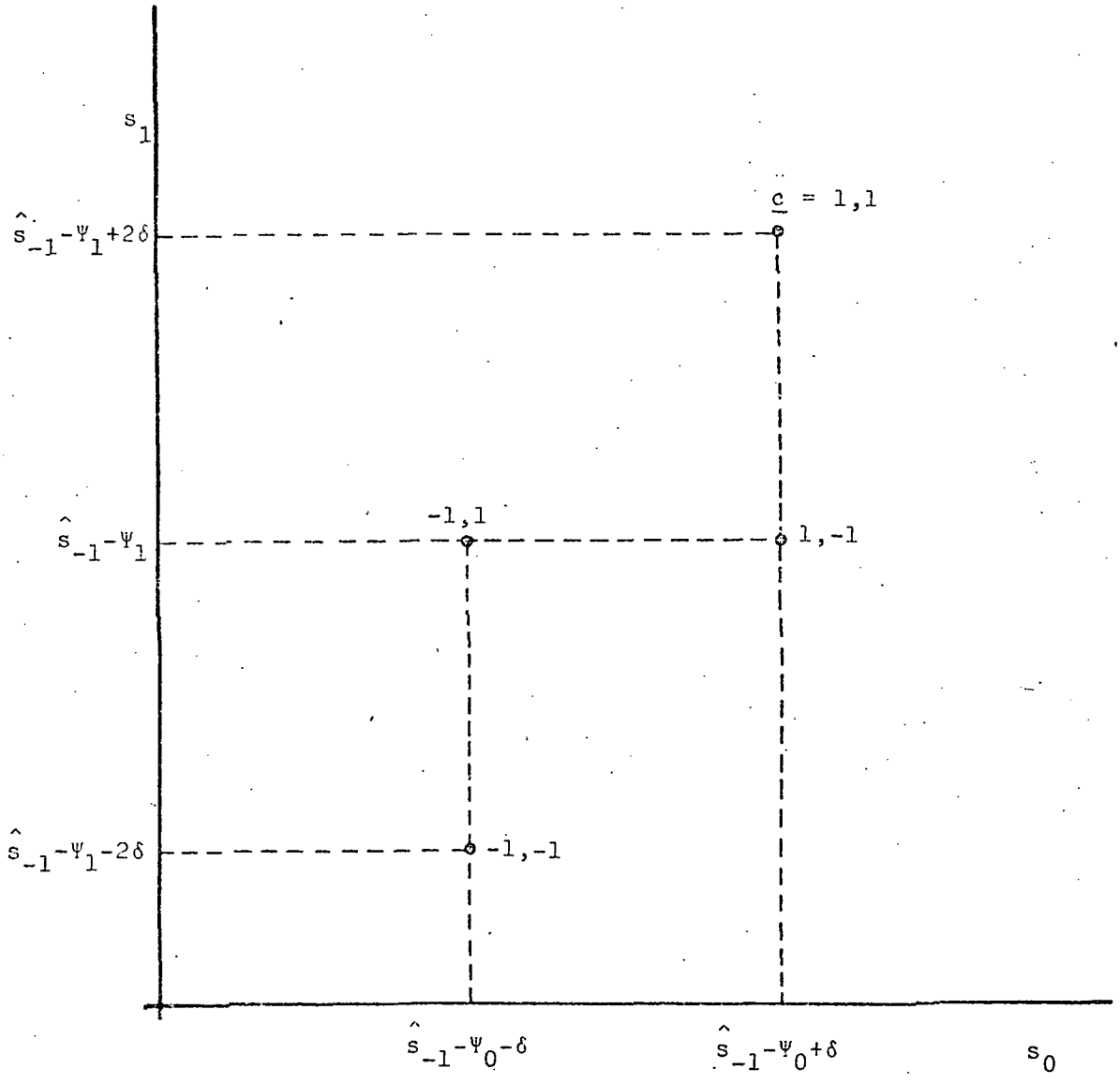



FIGURE 4.1
BLOCK-2 DM ENCODING PLANE
SHOWING $\vec{S}_f$opt POINTS

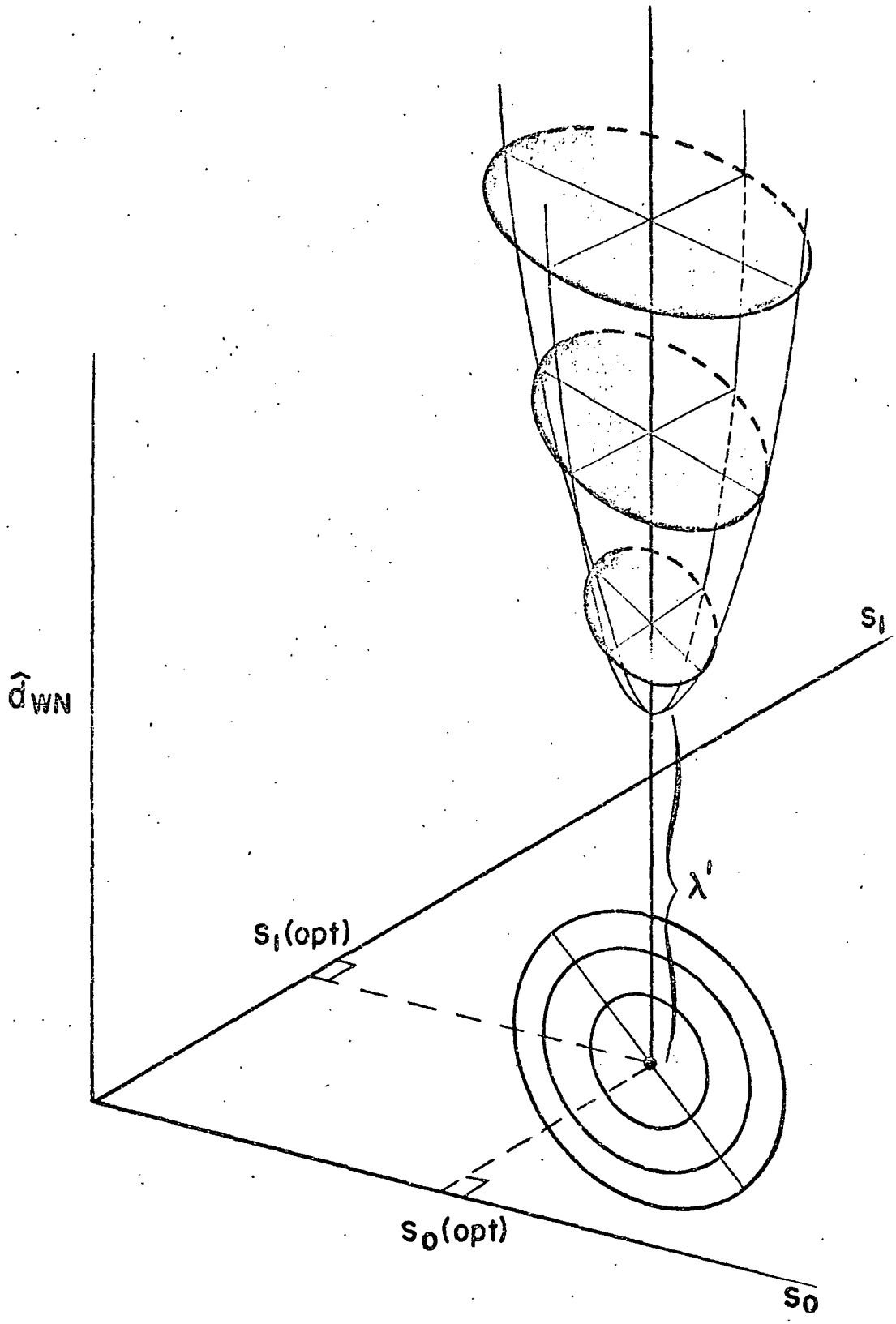



FIG. 4.2

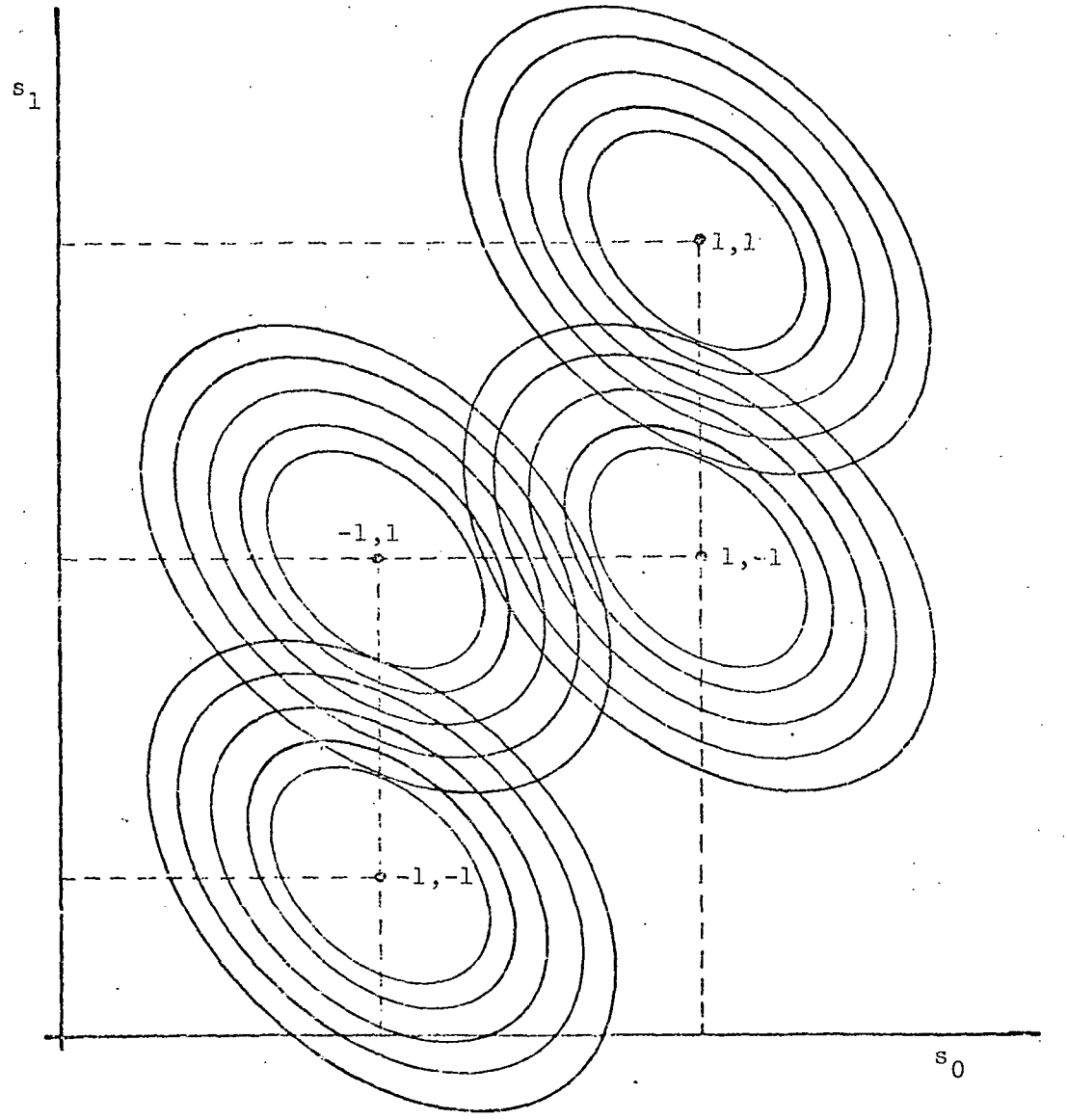



FIGURE 4.3
BLOCK-2 DM ENCODING PLANE
WITH EQUAL $\hat{d}_{WN}$ FAMILIES

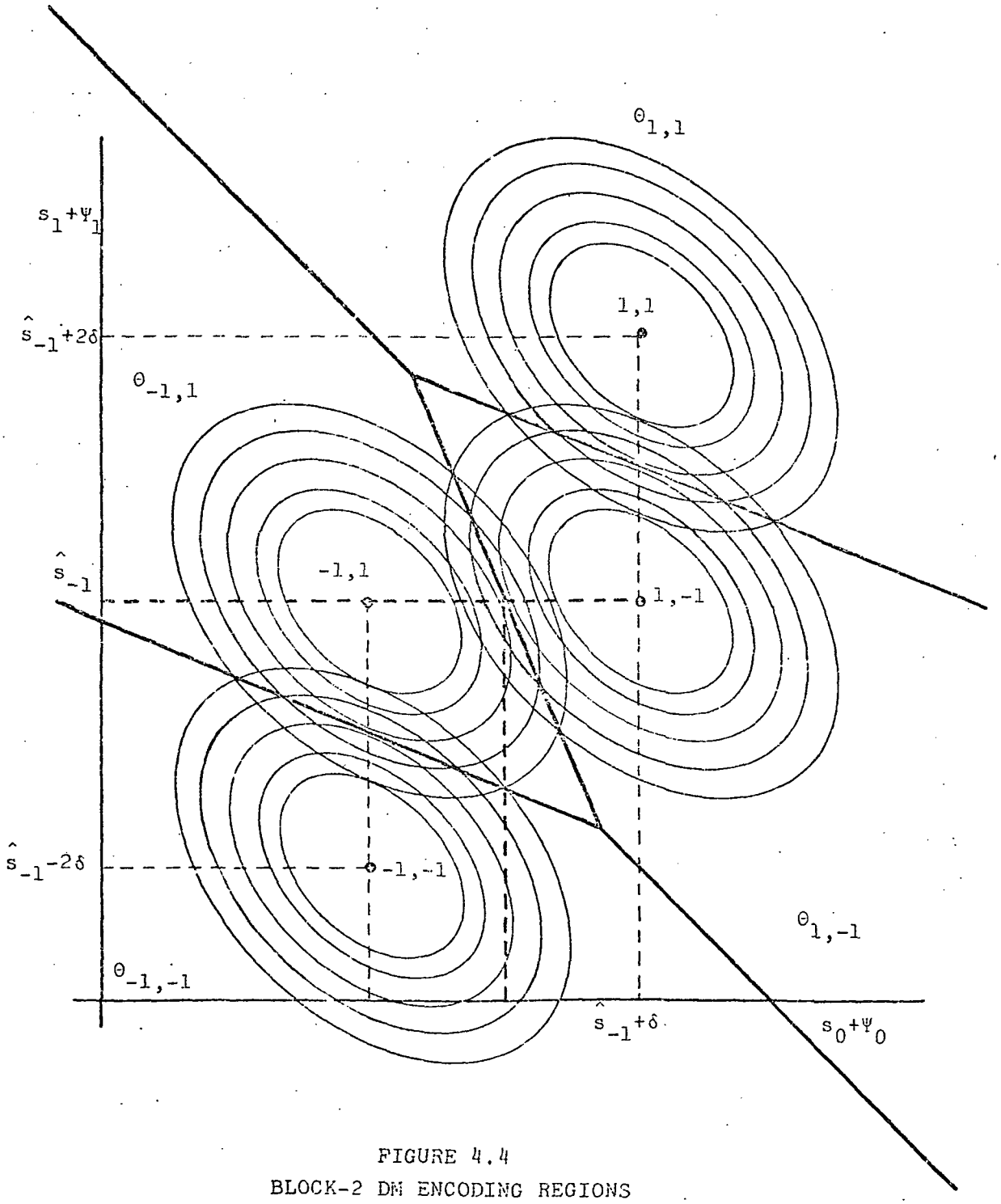



FIGURE 4.4
BLOCK-2 DM ENCODING REGIONS

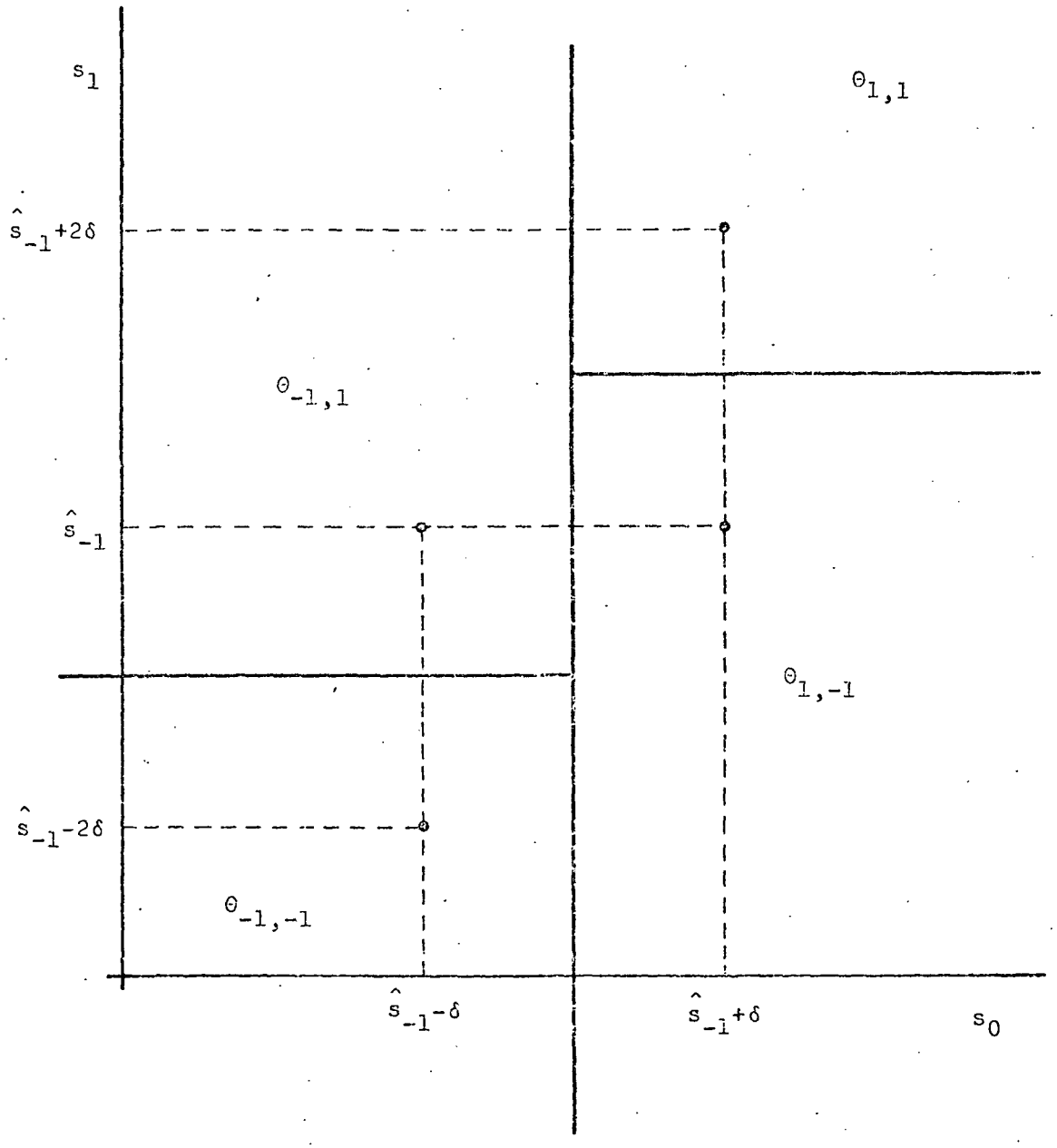



FIGURE 4.5
STANDARD DM ENCODING REGIONS IN TWO DIMENSIONAL SOURCE SAMPLE SPACE

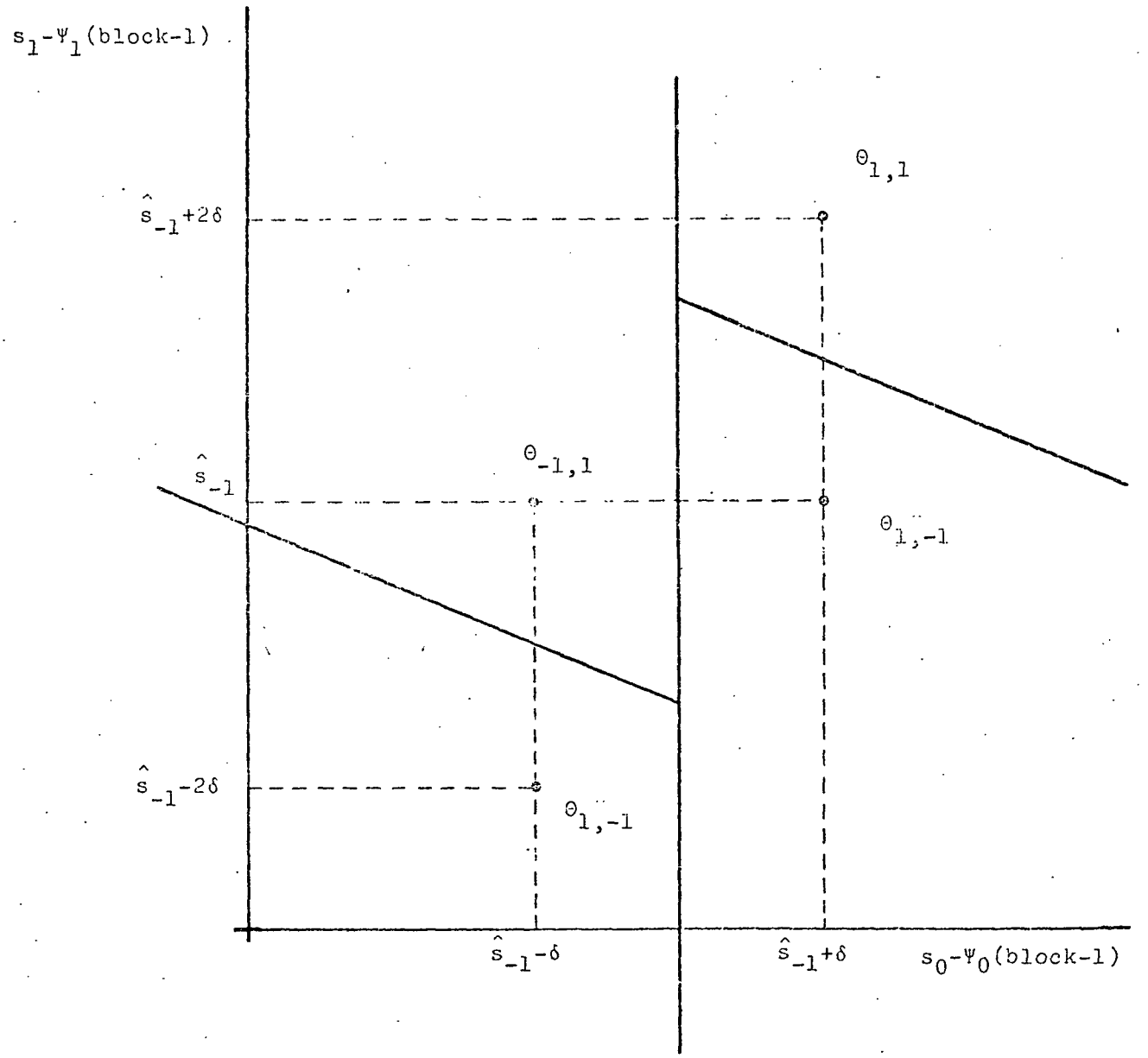



FIGURE 4.6

BLOCK-1 DM ENCODING REGIONS IN TWO DIMENSIONAL SOURCE SAMPLE SPACE

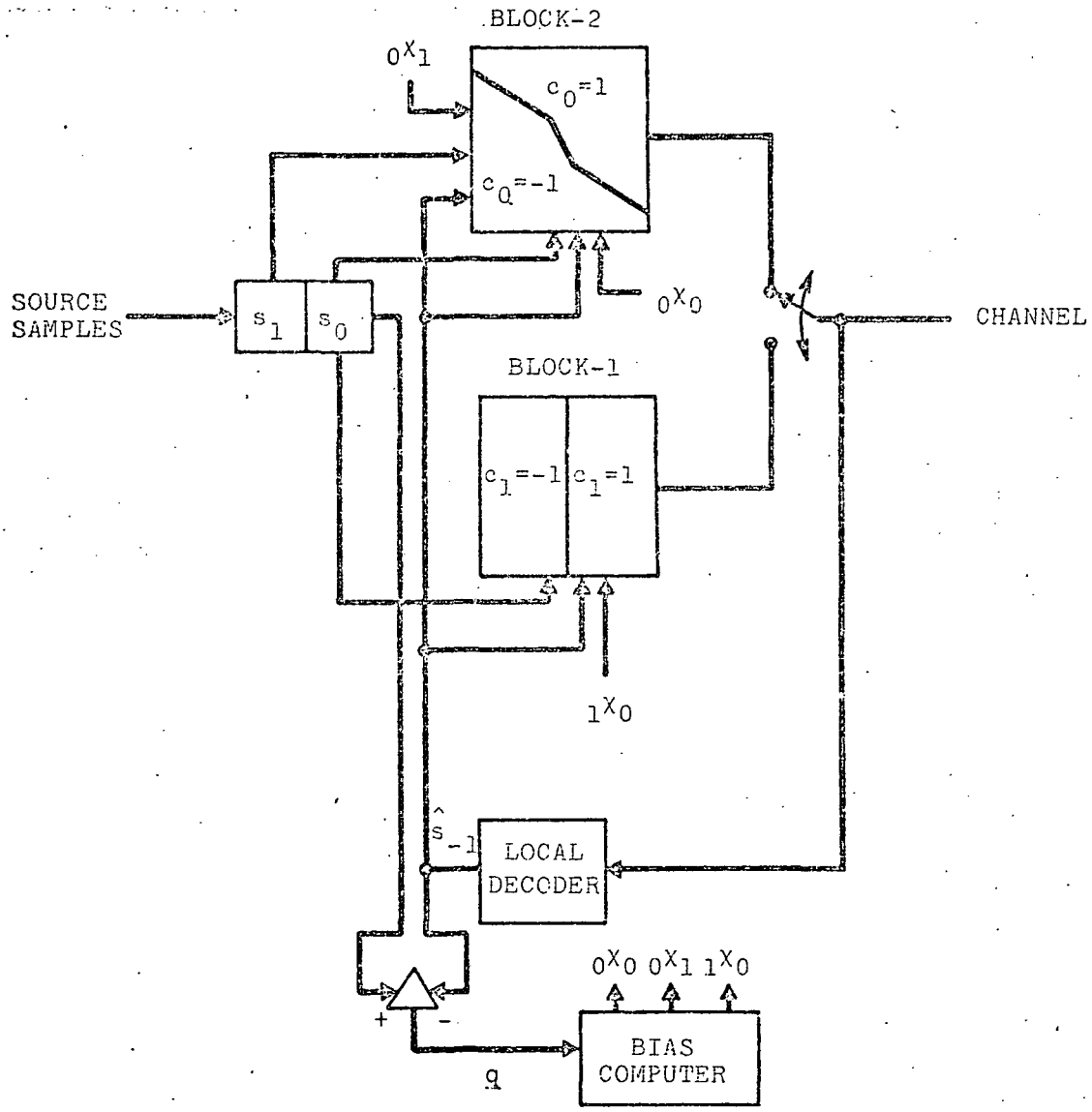



FIGURE 4.7
STREAM MODE BLOCK-2 ENCODER

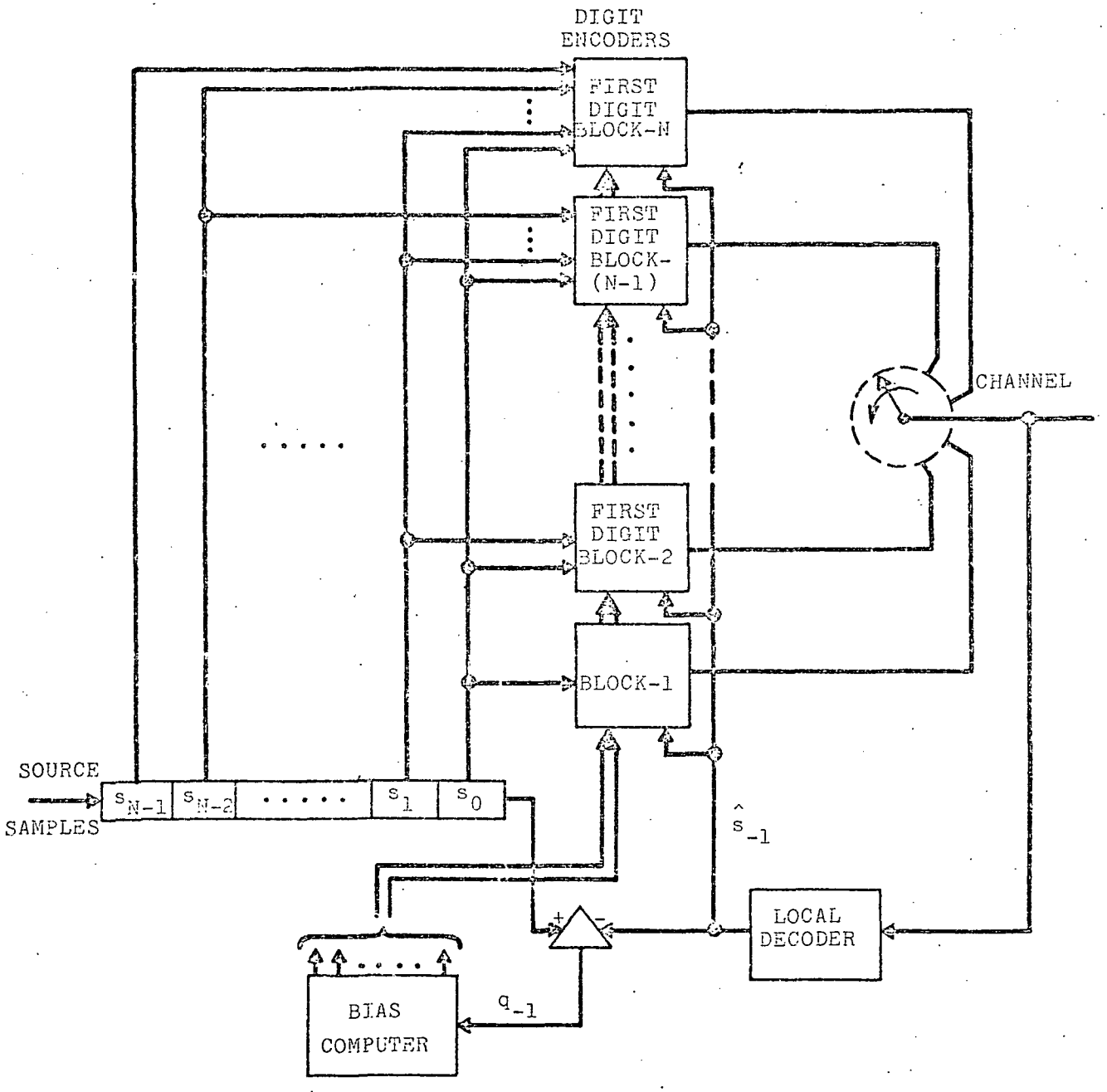



FIGURE 4.8

STREAM MODE BLOCK-N ENCODER

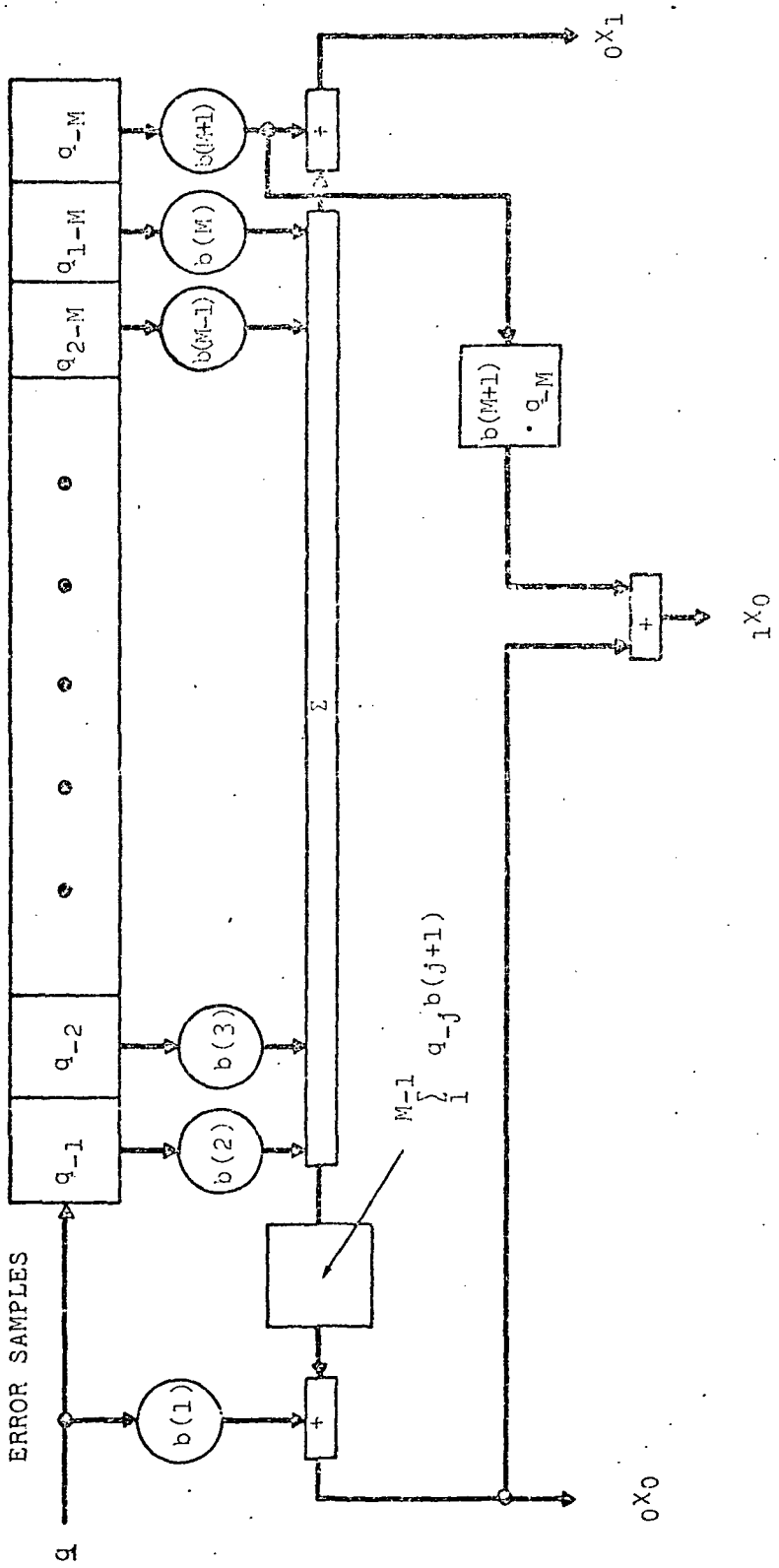



FIGURE 4.9

BIAS COMPUTER FOR BLOCK-2 STREAM MODE ENCODER

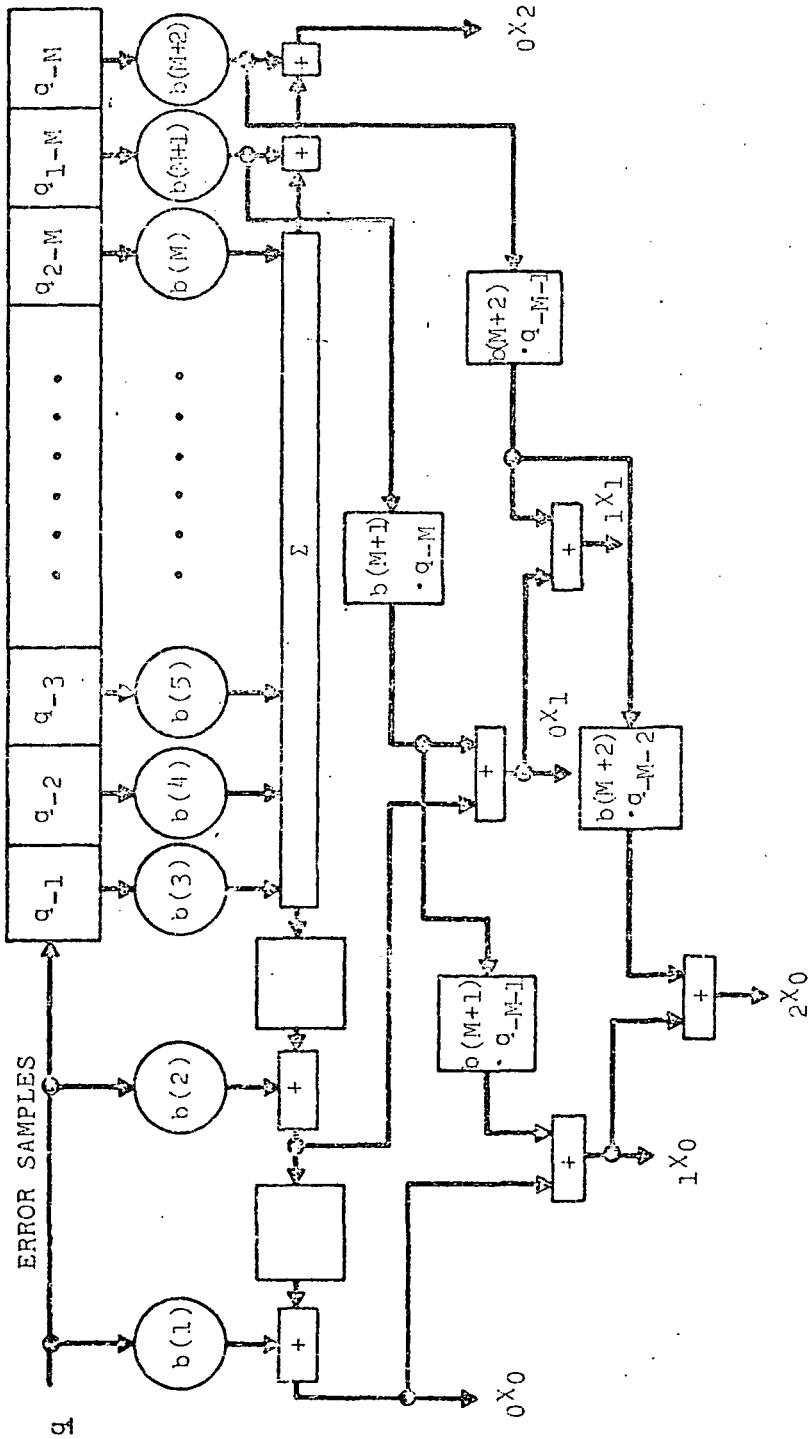



FIGURE 4.10
BIAS COMPUTER FOR BLOCK-3 STREAM MODE ENCODER

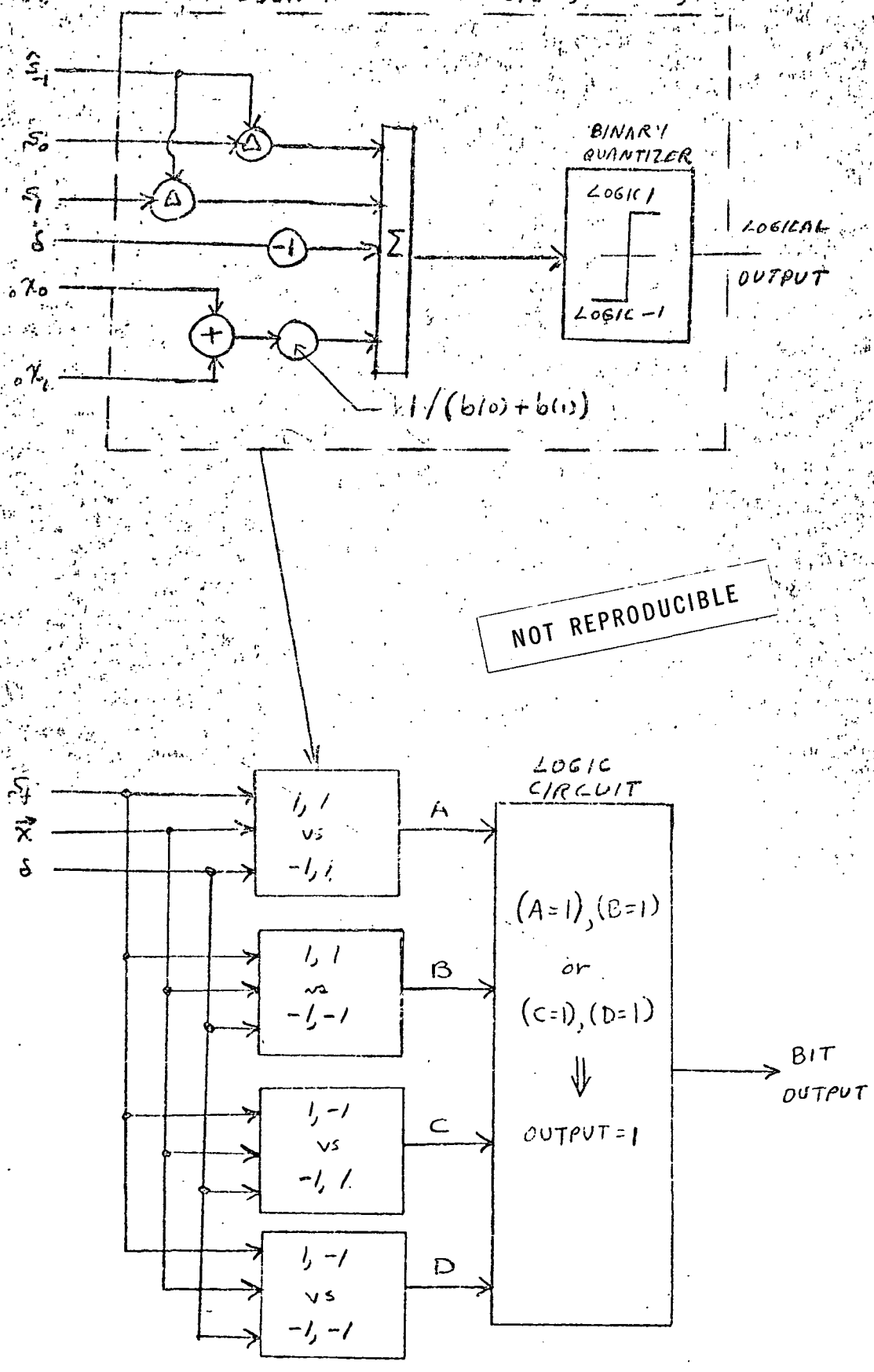



FIG. 4.11

FIRST DIGIT UNIT FOR BLOCK-2 DM
STREAM MODE ENCODER

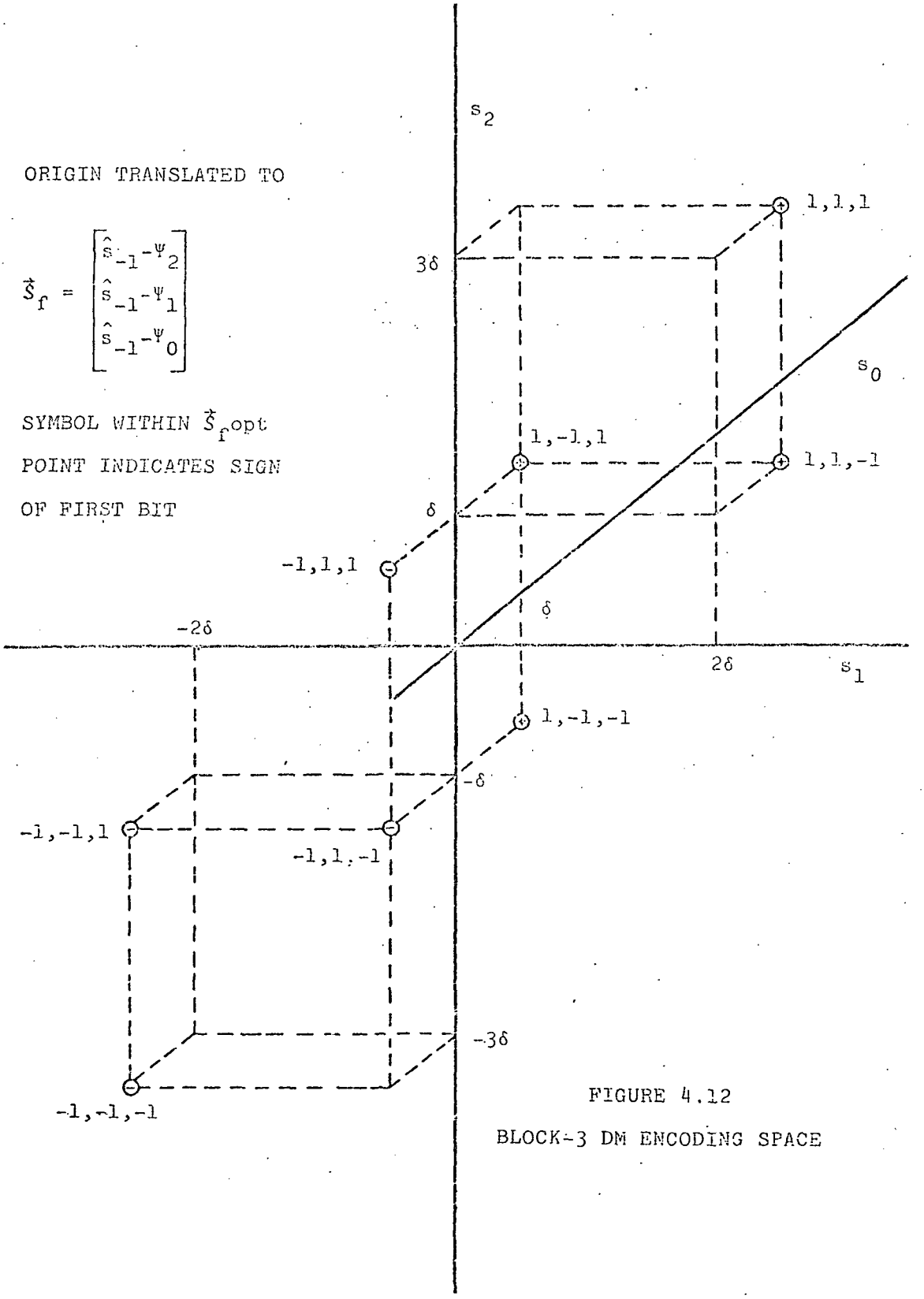



FIGURE 4.12

BLOCK-3 DM ENCODING SPACE

# Chapter V

## 5.1 Outline of the Simulation Work

The experimental part of this research consists of a digital computer simulation study of the weighted noise encoder as applied to linear, single integration DM. Short sentences of speech serve as the main source material, in keeping with a test which is as meaningful as possible. Performance is judged quantitatively on the basis of the frequency content of the encoding noise, as determined by subjecting the error sequence to spectrum analysis, and qualitatively by having the reconstruction sequence converted to a low pass analog signal for audition. Although no formal program of subjective testing was undertaken, listening to the output provided definite corroboration of the results seen in the noise spectral density output.

The motivation for a computer simulation study, in a field where the proof of the pudding usually rests in subjective tests of the actual device, was the availability of certain exceptional software and hardware research tools at Bell Laboratories. An IBM System 360 computer, with an extensive computer aided graphics package, was an important component. The complete chain from analog input to analog output was made possible, however, by an ultra high fidelity digital recording system (DRS) which samples the source, records the samples on a 360 compatible digital blocked tape, and plays back (a computer generated) blocked

tape to convert the samples to a continuous output for audition, with virtually negligible noise or distortion of its own.

A complete experimental run entails the following general steps. First, the source material is recorded on DRS, creating an 8 track blocked tape which mounts directly on the IBM 360 tape drive. The simulation program reads this input tape and converts from the special DRS format to a sequence of source sample values. The source sample sequence is then processed by the encoder simulator, and the local decoder output sequence is put back into DRS format and written on an output tape. Any processing of the error sequence (eg, spectrum analysis) can be done concurrently since the operation is not in "real time". After the run is completed, the output reel may be taken and replaced on the DRS tape drive for playback.

The encoder simulation itself is relatively a very small part of the entire programming effort. Reading and writing of tapes, and the associated format conversions, the digital filtering necessary to effect sampling rate multiplication and output interpolation, the spectrum analysis and graphical output routines, and the supervisory programs were all more significant problems. A full discussion of the programming, as well as of related theoretical questions, is reserved for Chapter 6. The remaining sections of this chapter are devoted to the simulation results, but a few key facts regarding the data processing are required beforehand.

The speech material, in the form of a short utterance such as "the boy was there when the sun rose", was sharply band limited to below 4 kHz before being sampled at 8 kHz. For the above sentence, this yields approximately 25,000 Nyquist rate samples, each sample being given a 14 bit linearly companded representation by RDS (ie, quantized into one of $2^{14}$ values with some piecewise linear compression characteristic). The Nyquist samples were generated and stored in blocks (logical record length) of 512.

An effective sampling rate of 64 kHz, or 8 times Nyquist, was accomplished by applying an interpolating filter to the Nyquist samples, which fits a low pass function through the original points in order to recover the seven interstitial values. Thus the record length for DM simulation operating at 64 kHz is 8 x 512 = 4096 samples. Submultiple (of two) speeds 32 kHz and 16 kHz, were easily simulated by deleting samples appropriately in the 64 kHz records.

The encoder simulation program processes record by record, generating 4096 element arrays of reconstruction and error samples. The DRS cannot reproduce 64 kHz samples directly because of hardware limitations. It was necessary to collapse the high rate output values back to an 8 kHz sequence* by a low pass filtering process.

---

* For the reconstruction the sampling rate cannot be divided by simply taking every eighth member of the sequence. This is explained in Chapter 6.

The error records were retained for power spectrum estimation using the discrete fourier transform, implemented as an FFT, in a modified periodogram technique. A data window is applied to the 4096 values before the FFT is computed. The squared magnitude of the resisting transform, in some cases averaged over several records, provides the power spectral density estimate at 2049 discrete frequencies from dc through $\frac{1}{2\tau}$ (the folding frequency), at uniform intervals of roughly 15 Hz.

The computer generated plots display the spectral density in decibels vs. linear frequency, the data being connected sequentially by straight lines rather than showing just the individual points. Because of the great density of data, this method gives the appearance of a histogram, although such was not intended. In fact, if certain frequencies dominate, ie, surrounding estimates are much smaller as in the case of periodic time domain data, the plot looks very much like a "line spectrum." It must be remembered, however, that a "line" does not necessarily signify the presence of a sinusoidal component in the original data at precisely that frequency, or any of its aliases. Rather, the interpretation is that is is an unbiased estimate of the true power spectrum at that discrete frequency, when viewed through the distortion of the appropriate spectral window. Nevertheless, a strong isolated point does indicate a periodic component near it in frequency because the spectral window half power width is of the order of the transform variable spacing, 15 Hz, and the first side lobes of the window

are down about 80 dB from the central peak. This means that leak-through is negligible, while resolution remains adequate for these purposes.

## 5.2 Sinusoidal Input

The sine wave simulation tests were conducted with a 3.5 kHz, $2\sqrt{2}$ volts p-p input, which is 6 dB below slope overload of the 64 kHz DM operating at a step size of 1 volt.* Although the block-N encoder performs best when there is a comfortable margin against overload, these results still reflect most of the properties of minimum $d_{WN}$ encoding seen in the speech input simulation results to follow later. For a reference, Figure 5.1a shows a standard DM encoding of the sine wave. In this tracking diagram, the asterisks are the local decoder outputs at the operation instants, of which about 50 are shown. The instantaneous differences, ie, the error sequence, is also given, with the zero error line arbitrarily relocated to -6 volts. One may verify upon inspection the standard DM rule: whenever the input at a sample instant exceeds the previous reconstruction (asterisk), the present reconstruction is one step up, and vice versa.

A spectrum analysis of the error sequence, Figure 5.1b, shows that the noise energy is concentrated in narrow lines, in this case 1 kHz apart. The discreteness in frequency means the error sequence is approximately periodic, repeating, with sparse exception every 64 samples. This is not too surprising, when considering the DM encoder as a (highly) nonlinear sampled data system with a sinusoidal input. In an oversimplified

---

* The maximum signal slope, $2\pi\sqrt{2}\cdot 3500$ volts/sec ($\approx$ 31,000) is about half the overload value $\delta/\tau$ = 64,000 v/sec.

view, the nonlinearity (of the binary quantizer) generates harmonics of the 3.5 kHz input, which are folded back, or aliased, into the baseband because of the sampling. The component at 2.5 kHz, for example, could have resulted from the 19th harmonic, 66.5 kHz, which is its alias.

This power spectrum is consistent with that found both experimentally and analytically in previous work[14]. It has, as a gross judgment, a fairly flat characteristic. If the source band is 0 to 4 kHz, there are four in-band noise components, including one at the signal frequency. It may not be fair to label that one as noise, however, since at worst it represents phase shift, and/or attenuation of the signal frequency component in the output relative to the input.

Consider next the identical input and decoder, but using the block-1 encoder. With unity in-band weighting,* and M=24, the tracking is as shown in Figure 5.2a. Although superficially little or no different from the standard DM tracking, the effect of noise weighting on the encoder decisions is discernible. The arrows point to instances where the block-1 reconstruction departs from that of standard DM.† Because standard DM is minimum difference encoding, these

---

* The ostensible weighting is 1 in-band and zero outside. Recall, though, that the effective noise weighting is this this ideal characteristic convolved with the spectral window $V(\omega)$.

† Each isolated departure represents two successive different decisions. The first causes the reconstructions to diverge, and the second reunites them.

points of departure must represent errors which are larger, hence raising the MSE.

The block-1 decisions are justified, however, by the spectral analysis of the noise in Figure 5.2b. The line at the signal frequency has been reduced about 9 dB, but more significantly the second component (1.5 kHz) has been suppressed 8 dB, and the third (2.5 kHz) has been all but eliminated. This, of course, is at the expense of higher out of band noise. That the total spectrum is larger, reflecting the increased MSE, can easily be seen. Incidentally, the fifth noise line (4.5 kHz) has also been reduced by block-1 encoding, even though it is out of the signal band. The explanation is that the effective weighting function, due to the spectral window, is still large enough at that frequency to cause suppression of this noise component.

The block-3 encoder decisions are shown in Figure 5.3a. Here the increase in MSE over standard DM is quite evident from the dispersion of the noise sequence, and there are accordingly more departures from the standard DM tracking than with block-1. There are enough, in fact, to erase the 64 sample quasi-periodicity, stretching the noise repetition interval threefold to about 192 samples. This is inferred from the closer spacing of lines in the noise power spectrum density, Figure 5.2b.

Now the signal frequency line is reduced to -39 dB, a substantial improvement that is attributed to the look-ahead

property of a large block size which allows the reconstruction to follow with, in this case, extremely small phase and amplitude error.

Another characteristic seen in the larger block size encoders is reduction of flat weighted noise by control of all the in-band components to produce a more or less flat in-band spectrum, as contrasted with a large reduction of just some of the in-band components obtained with block-1. Of course, results must still be metered by the chosen criterion, which in the present case is estimated by a straight summation of the squared magnitude of the noise components up to 4 kHz. Going beyond that, however, one can see an advantage on a subjective basis to having discrete noise components broken up and spread out into more of a continium on the frequency axis, as well as eliminating or converting any dominant noise components which would be distinctly audible.

## 5.3 Speech Input

The bulk of the simulation work was carried out using a catalog of six recorded utterances, three spoken by a male and three by a female. All were utilized for the informal listening tests, but the noise spectrum studies to be reported in this part concentrated on the 5th and 8th of the 48 data (source sample) records comprising the male sentence, "The boy was there when the sun rose." No special reason singled out these two records, except that a visual scan of the printout showing the 512 sample values in both indicated a fair amount of activity which was reasonably "stationary" throughout each record, with no bad surges or quiet areas.

The power spectra of these two records, when expanded to 4096 64kHz samples, is shown in Figure 5.4. The out-of-band energy has resulted from the necessarily imperfect bandlimited interpolation, but this spurious energy is small enough so that one may ignore the effects of signal aliasing on the quantizing noise results. The interpolation (sampling rate multiplication) problem, vis-a -vis the spectra in Figure 5.4, is taken up in detail in Chapter 6.

The dominant spectral components are seen to be around 800 Hz, which happens to be a favorite test tone frequency of DM investigators. Also, the general structure of the spectra for the two records is similar, however, record 8 is about 6 dB higher in overall level.

The first series of three figures compares block-1 (with a very short past error memory of 8) against standard DM, based on record 5. In each run the entire sentence was processed so that these interior records are not subject to any start-up transients.* The step size in Figure 5.5 was .07 volts, a very low value which resulted in the encoder being in slope overload much of the time.

The two noise spectrum plots are almost identical because while in overload block-1 decisions are generally no different from those of a standard DM. A most interesting feature of these noise spectra is that the in-band structure is practically the same as the signal spectrum itself. This can be explained heuristically by noting that during overload periods, the output becomes merely a ramp, not following any of the detail of the signal trajectory. The error sequence, hence its spectrum, would be expected to contain all or most of that detail, for the higher signal frequencies at least.

The lack of significant noise energy above the signal band is further evidence of slope overload. When tracking properly, the error sequence tends to alternate, producing energy at higher frequencies. While in slope

---

* This is only a problem if encoding begins with a record whose first samples are large, and s has been initialized to zero.

overload, however, the error does not alternate, and so out-of-band noise components are not generated in strength.

The situation at .17 volts step size is substantially different, as Figure 5.6 shows. Runs were made at .02 volt step size increments, and this value was observed to be near optimum for the standard DM encoder, and this (5th) record. As is well known, standard DM exhibits a relatively sharp peak in output signal-to-noise ratio at that value of step size which produces just the right balance of slope overload and granular quantizing noise.

The block-1 noise spectrum is noticeably different, indicating that it is beginning to now make a certain amount of opposite decisions. The in-band noise, however, is not judged to be significantly less than that of standard DM, on the basis of a casual inspection at least. At this step size, there is still a significant amount of slope overload time.

Doubling the step size produces a marked change in the comparison. Only 6 dB larger, $\delta = .43$ puts standard DM well into the granular noise region, and results in perhaps 10 dB higher noise than the optimum $\delta = .17$ results. On the other hand, the large step size has allowed block-1 to make favorable binary decisions, from an in-band noise standpoint, yet remaining out of slope overload danger. The in-band noise reduction is seen to be about 15 dB relative to standard DM at this step size. It appears that this

energy was pushed up into the middle part of the baseband, between 10 and 25 kHz.

Going higher in step size, one finds that the standard DM produces increasingly more granular noise, which empirically is always distributed with a bowl-shaped envelope similar to Figures 5.6a and 5.7a. On the other hand, block-1 maintains the in-band noise fairly constant over a broad range, certainly 10 to 12 dB in step size, with the additional granular noise being shifted out of band. Block-1 DM is therefore an inherently robust encoder in that the noise performance is effectively independent of the step size, conversely the input level, over a wide range above slope overload.

A representative comparison of block-1 vs. block-2 is seen in Figure 5.8, which is the noise from record 8 with $\delta$ arbitrarily set to .33 volts. Again, the ostensible weighting is unity in-band, zero outside, and the block length L is kept to 25 in each case. This block-2 therefore trades one past error for an additional source sample.

The major effect of the block-2 decisions is a much diminished phase and amplitude error in the output, so that the noise at the signal components (ie, noise which would have a high cross spectrum with the signal) is sharply reduced. As observed in the sine wave input comparisons between block-1 and block-3, the result is an in-band noise spectrum which has been made flatter and

much less signal dependent. This may be an important factor not registered by the frequency weighted error criterion, as signal dependent noise often has a raspy quality which can be more objectionable than independent flat noise.

Figure 5.9 is a comparison between the previous block-1, record 8 noise and another block-1 encoder employing a single memory recursive filter which attempts to approximate a long nonrecursive filter. The details are given in Chapter 6, but suffice to say that it is equivalent to an infinitely long nonrecursive filter with tap gains

$$\frac{b(j)}{b(0)} = \left(\frac{3}{4}\right)^j \qquad (5.3.1)$$

It so happens that the ostensible $W(\omega)$ corresponding to this geometric series feedback weights is positive and monotone decreasing from DC to the folding frequency, and so some in-band noise reduction is expected. Indeed, the noise spectrum in Figure 5.9b shows surprisingly good in-band suppression, except at the band edge near 4 kHz. The reason is the very slow drop-off of this $W(\omega)$ at the band edge. It is suggested that this is also the

performance of the second integrator DM (Section 3.6), since the analog RC network has the sampled impulse response

$$e^{-j\frac{\tau}{\tau'}} = \left(e^{-\frac{\tau}{\tau'}}\right)^{j}$$

where $\tau'$ is the RC time constant.*

Finally, an example of long term averaged quantizing noise spectra is offered in Figure 5.10, comparing standard DM with block-1. The noise over about half the active speech time in the sentence has been averaged from individual record spectra. The reduced variance of the spectrum estimates resulting from the averaging is seen in the graphs.

The step size was adjusted to 1 volt so as not to slope overload on the peak energy records. This, however, puts the step size far into the granular noise region for most of the records, and therefore the characteristic heavy granular noise spectrum in Figure 5.10. These empirical results are inconsistent with the generally accepted notion that the granular noise is relatively constant with frequency.

---

* This is an approximation to the impulse response of the circuit shown in Figure 3.6a which neglects the second corner.

The almost 20 dB improvement in the block-1 noise, Figure 5.10b, is partially a result of the robustness, whereby a very large step size may be used to protect against slope overload without incurring the otherwise very strong granular noise of standard DM encoding. Of course, an adaptive decoder which can dynamically adjust its step size would not produce nearly as much total (ie, granular + slope overload) noise, but the present in-band improvement has been accomplished with the utter simplicity of the standard decoder.

The strong noise component at 16 kHz is unexplained. Since it only appeared in the 20 record block-1 average, it must be caused by block-1 forced periodicity in the error sequence in some brief segment of the sentence not examined on a record by record basis. However, it is well out of the signal band and presents no problem. It is conjectured that it happens during quiet periods, lapses between words, etc.

This is related to a practical problem of DM known as idle channel noise, wherein imperfect local decoders and binary quantizers can cause tones to appear in the signal band with no input present. Several remedies are available to counteract this phenomenon, however block-N encoding is a fortiori resistant to idle channel noise.

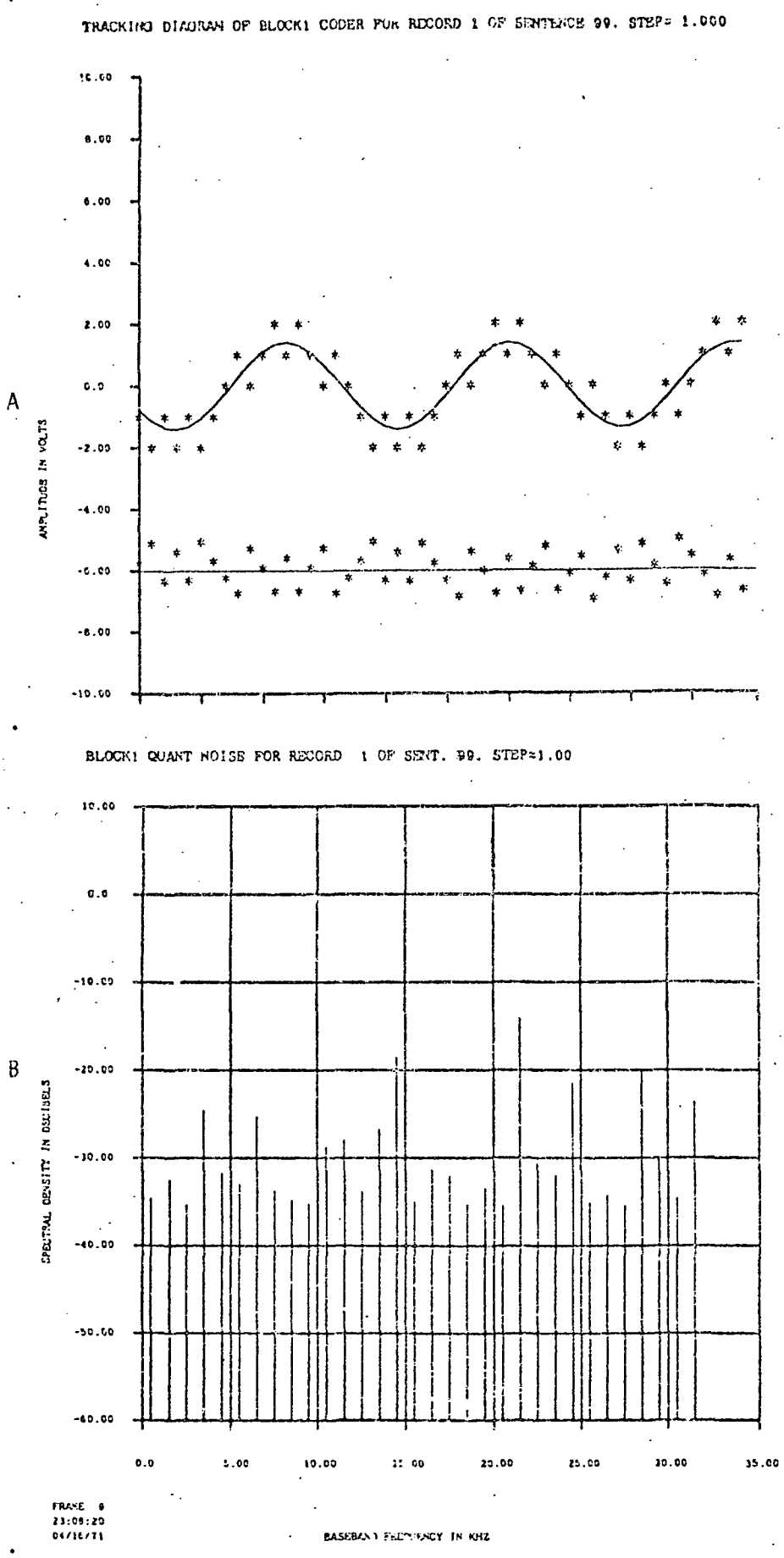



FIG. 5.1 STANDARD DM, SINUSOID INPUT, STEP=1.0
A. TRACKING DIAGRAM B. NOISE SPECTRUM

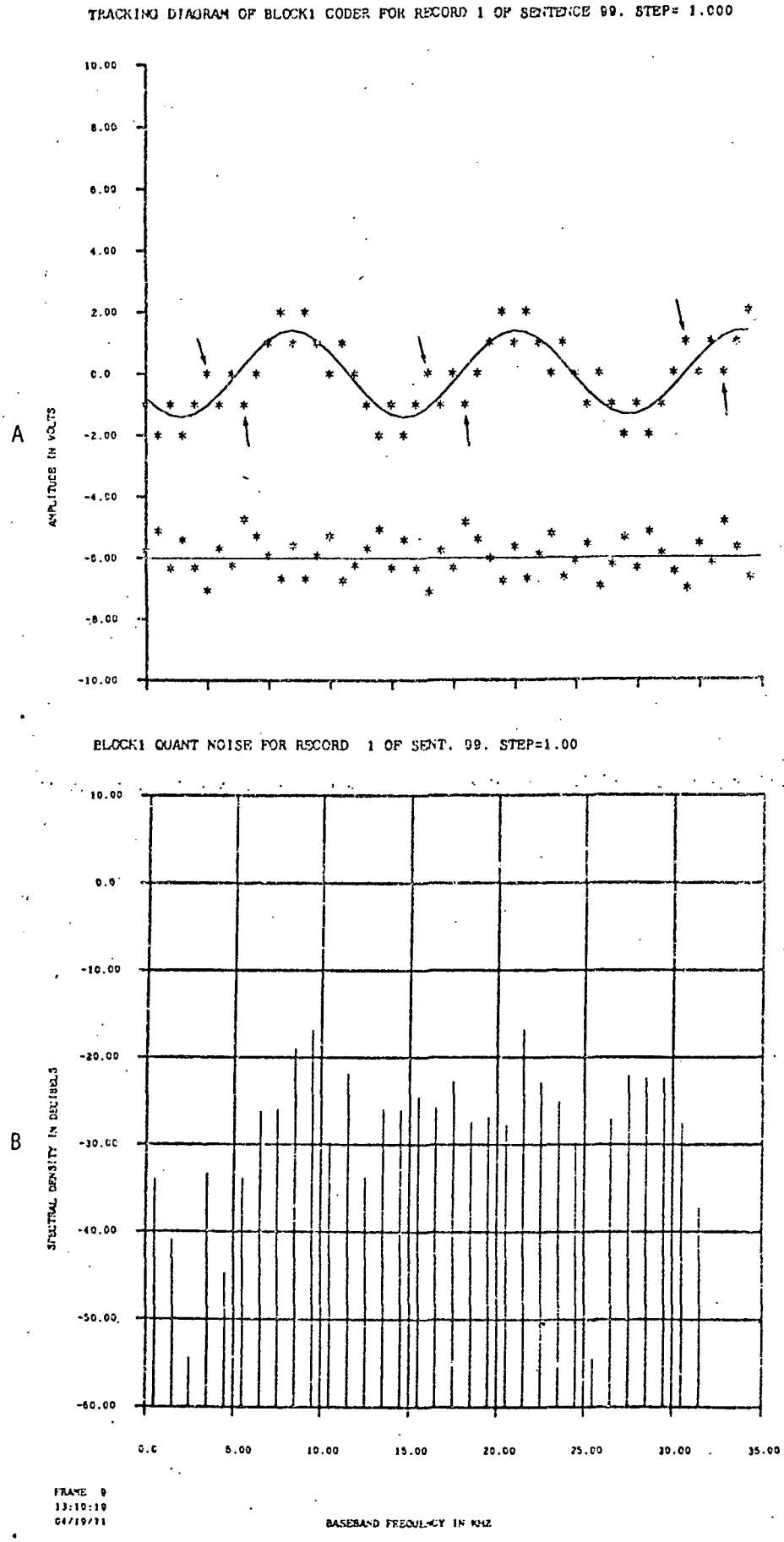



FIG. 5.2 BLOCK-1 DM, SINUSOID INPUT, STEP = 1.0
A. TRACKING DIAGRAM B. NOISE SPECTRUM

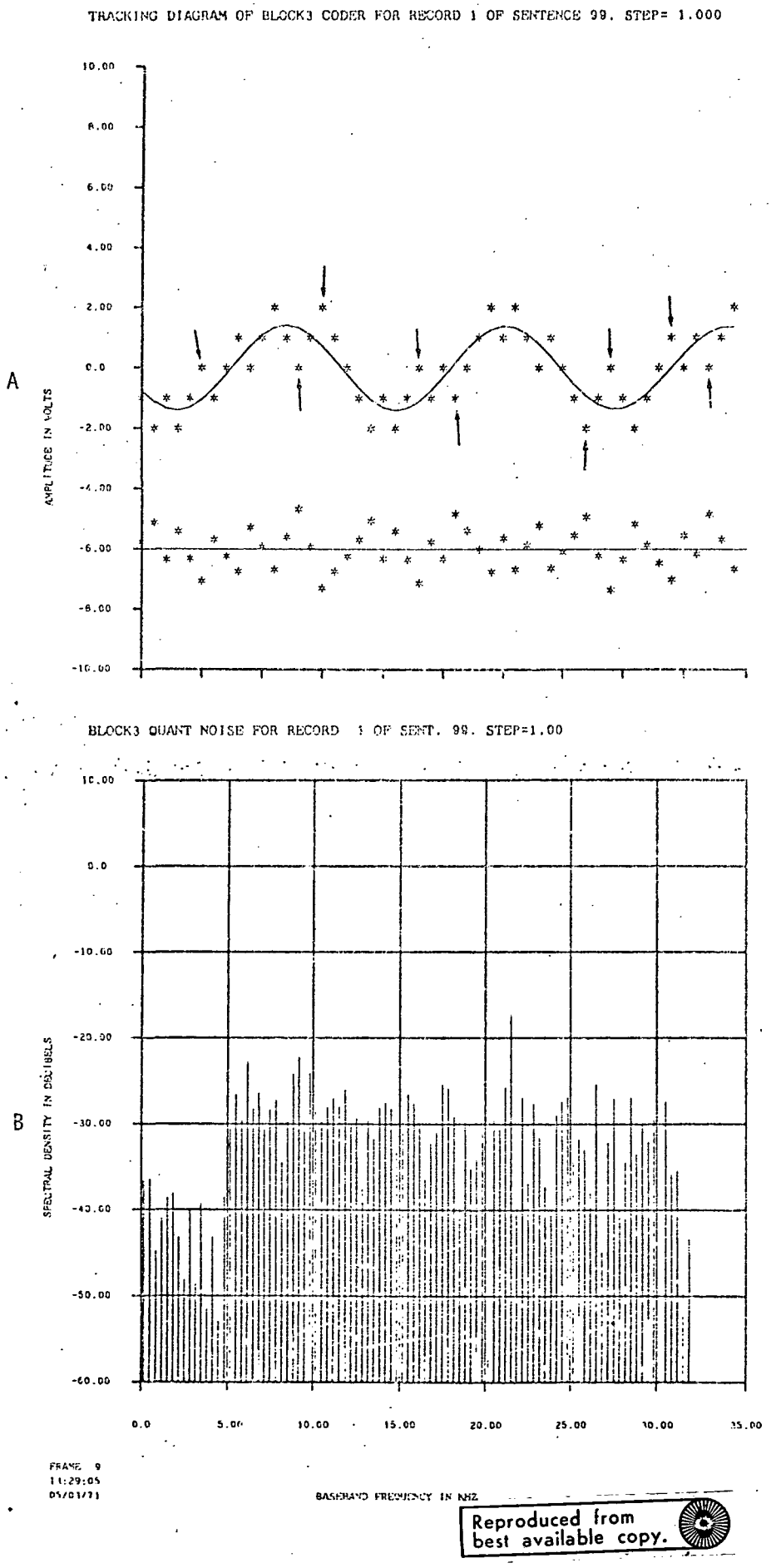





FIG. 5.3 BLOCK-3 DM, SINUSOID INPUT, STEP=1.0
A. TRACKING DIAGRAM B. NOISE SPECTRUM

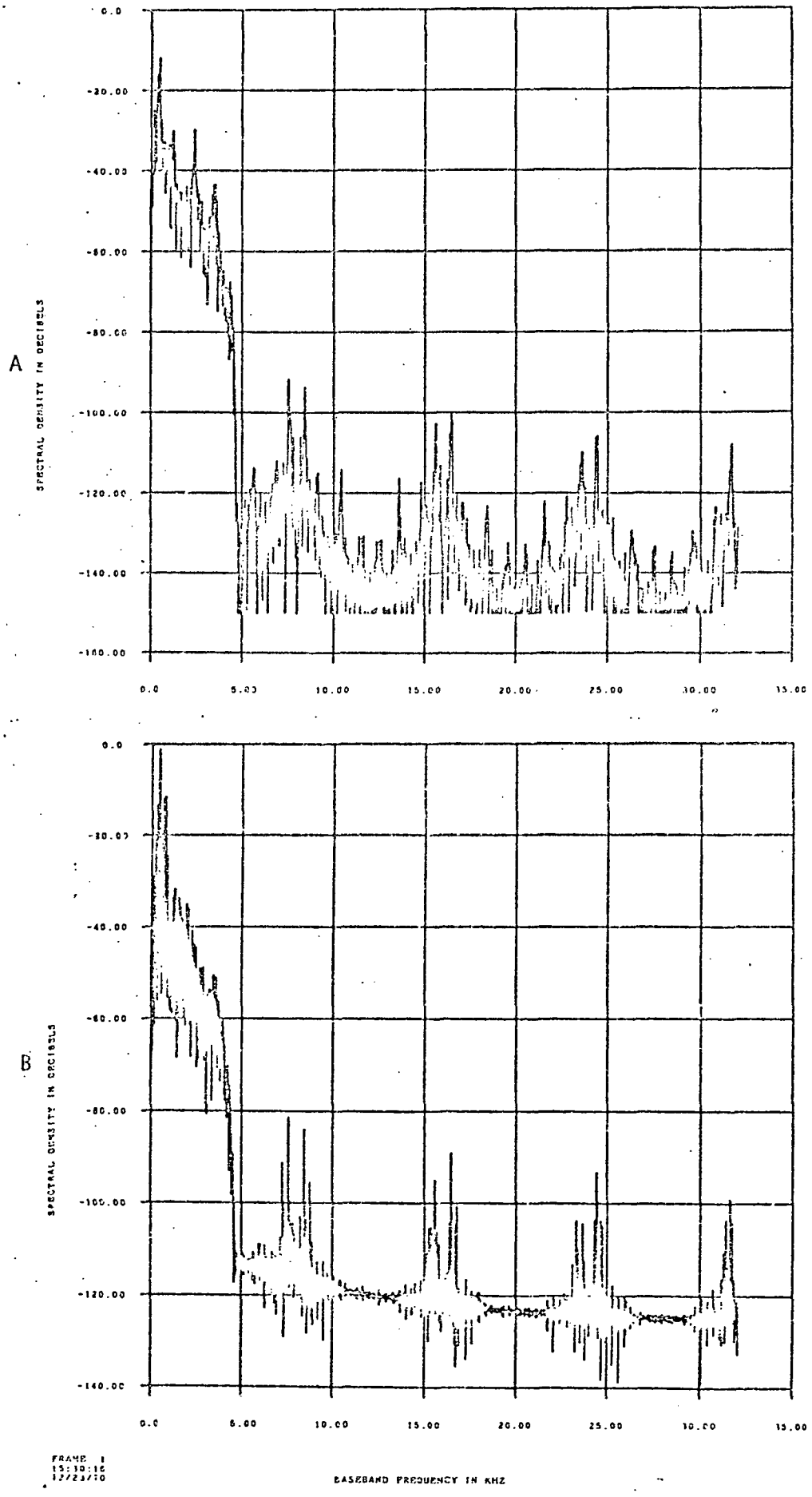
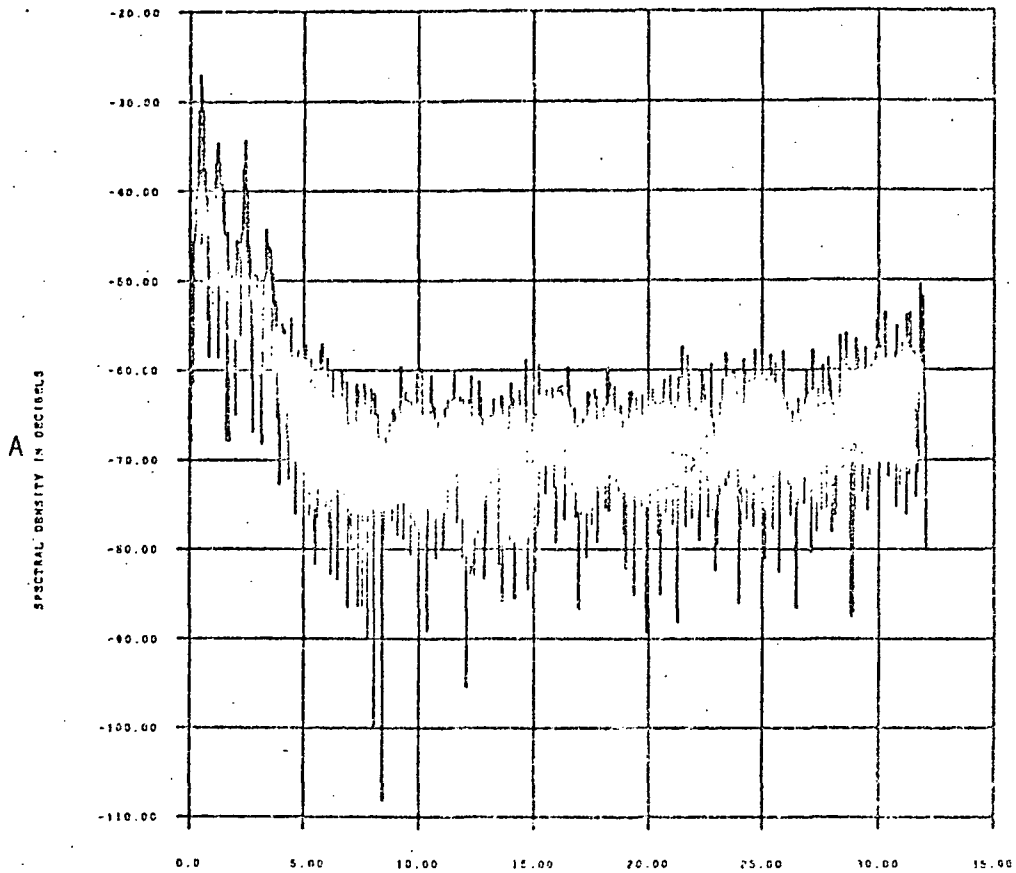



FIG. 5.4 POWER SPECTRUM OF INTERPOLATED 64 KHZ SPEECH SAMPLES

A. RECORD 5 B. RECORD 8

PERIODOGRAM OF NOISE WITH STANDARD DM 5 OF SENTENCE 1. STEP= 0.070




PERIODOGRAM OF SIGNAL OR NOISE FOR RECORD 5 OF SENTENCE 1. STEP= 0.070

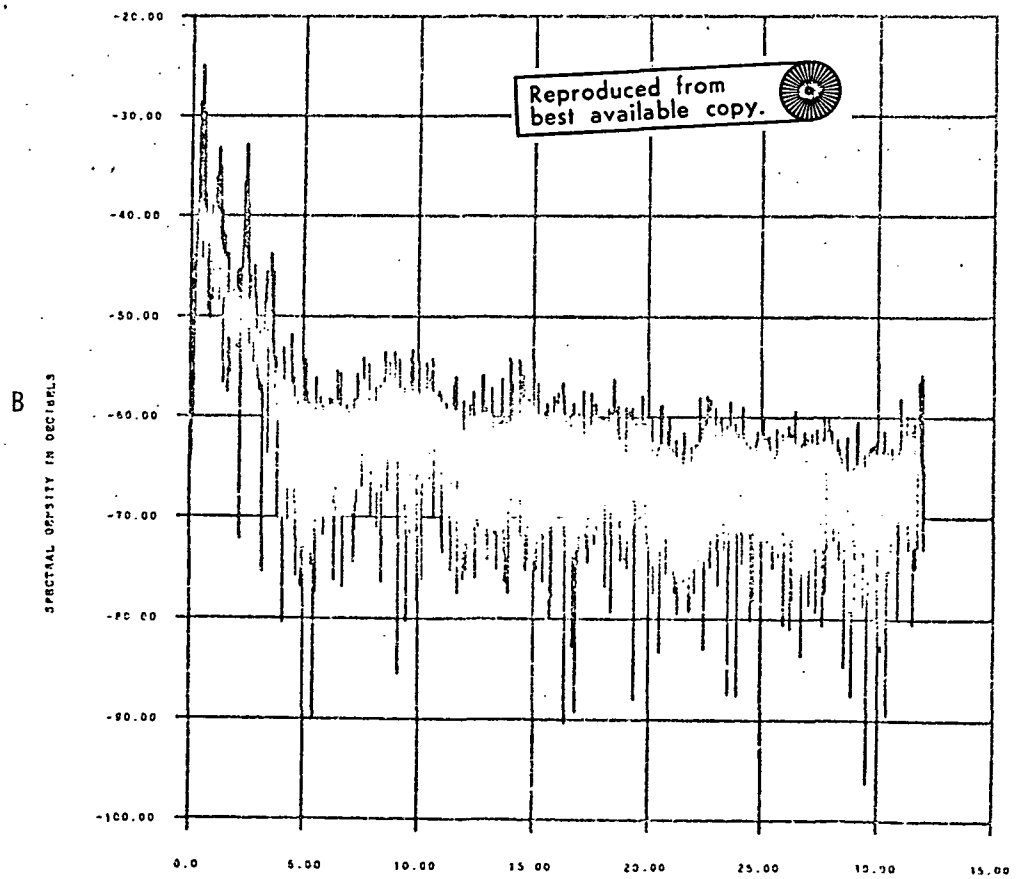
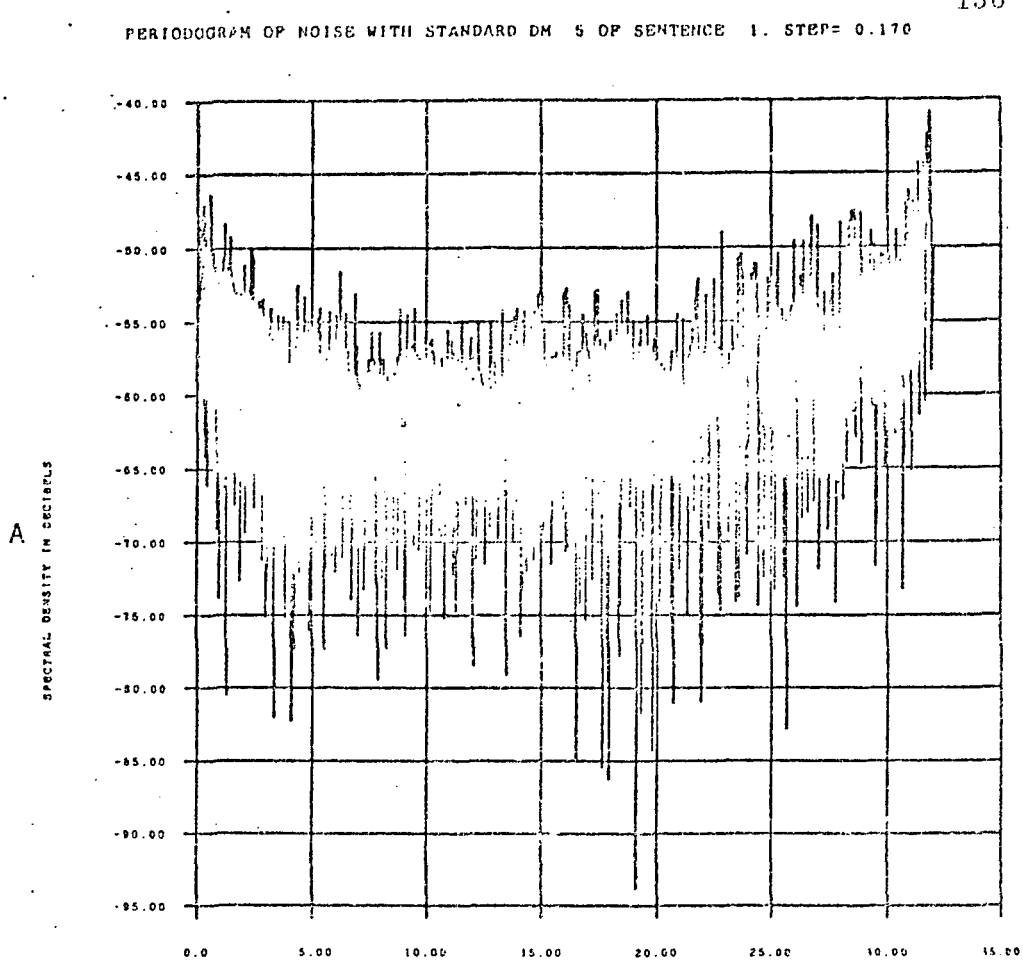





FIG. 5.5 NOISE SPECTRUM, RECORD 5, STEP=.07
A. STANDARD DM B. BLOCK-1 DM, M=8



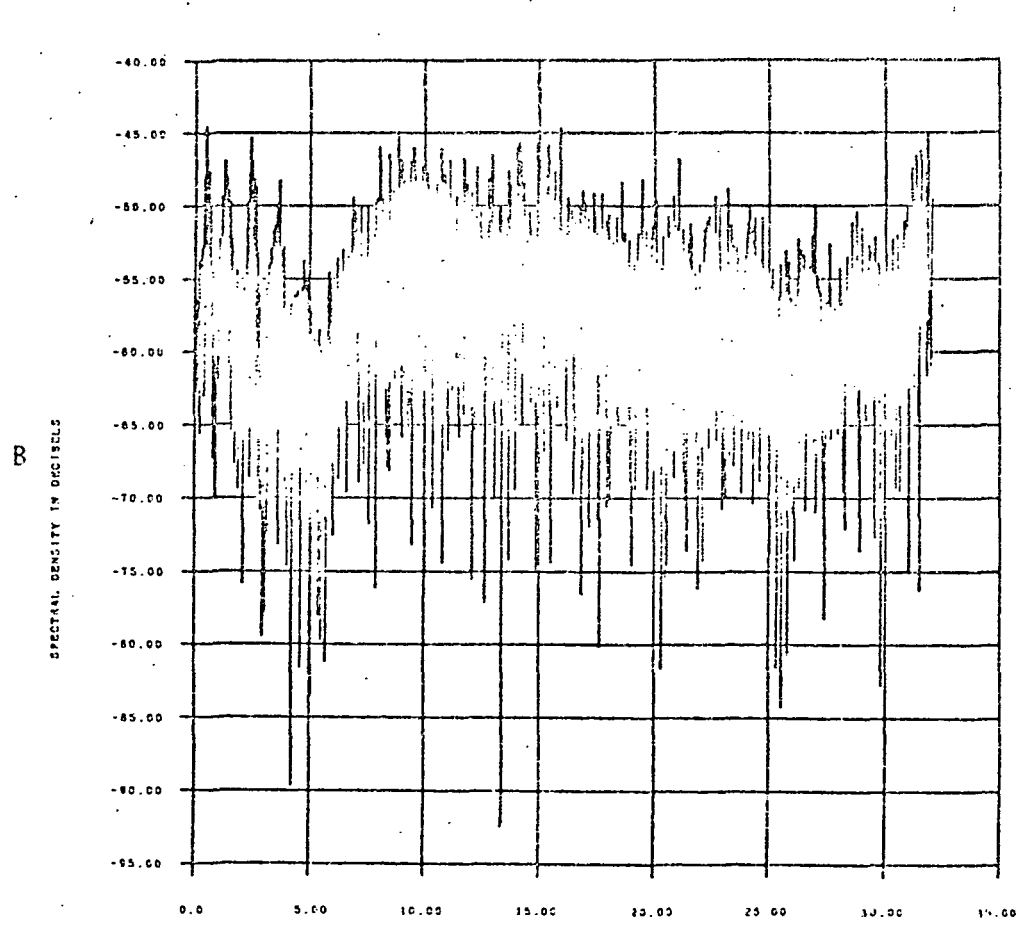
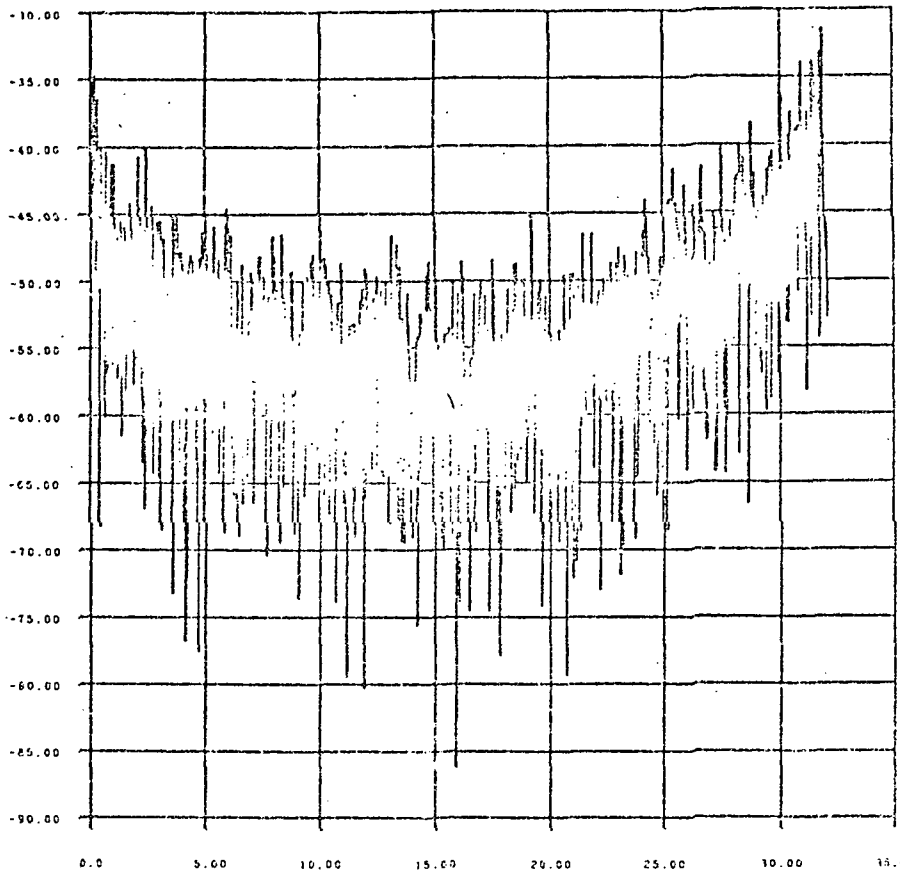




FIGURE 5.6 NOISE SPECTRUM, RECORD 5 STEP=.17

A. STANDARD DM B. BLOCK-1, M=8

PERIODOGRAM OF NOISE WITH STANDARD DM OF SENTENCE 1. STEP= 0.430




PERIODOGRAM OF SIGNAL OR NOISE FOR RECORD 5 OF SENTENCE 1. STEP= 0.430



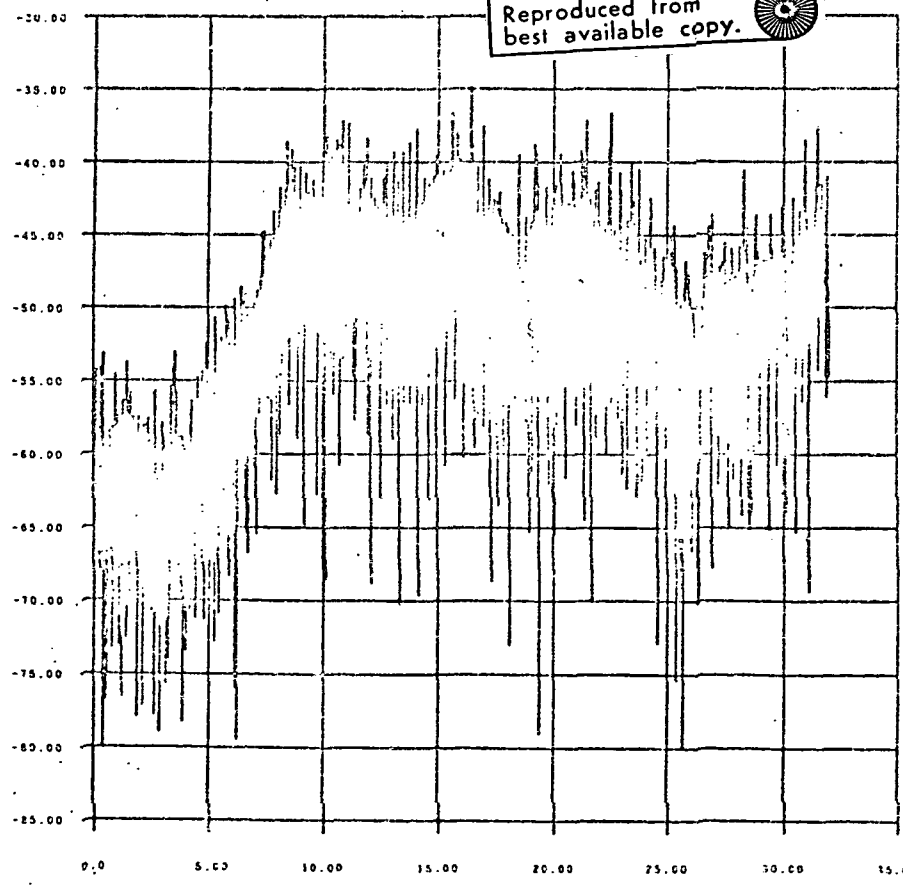
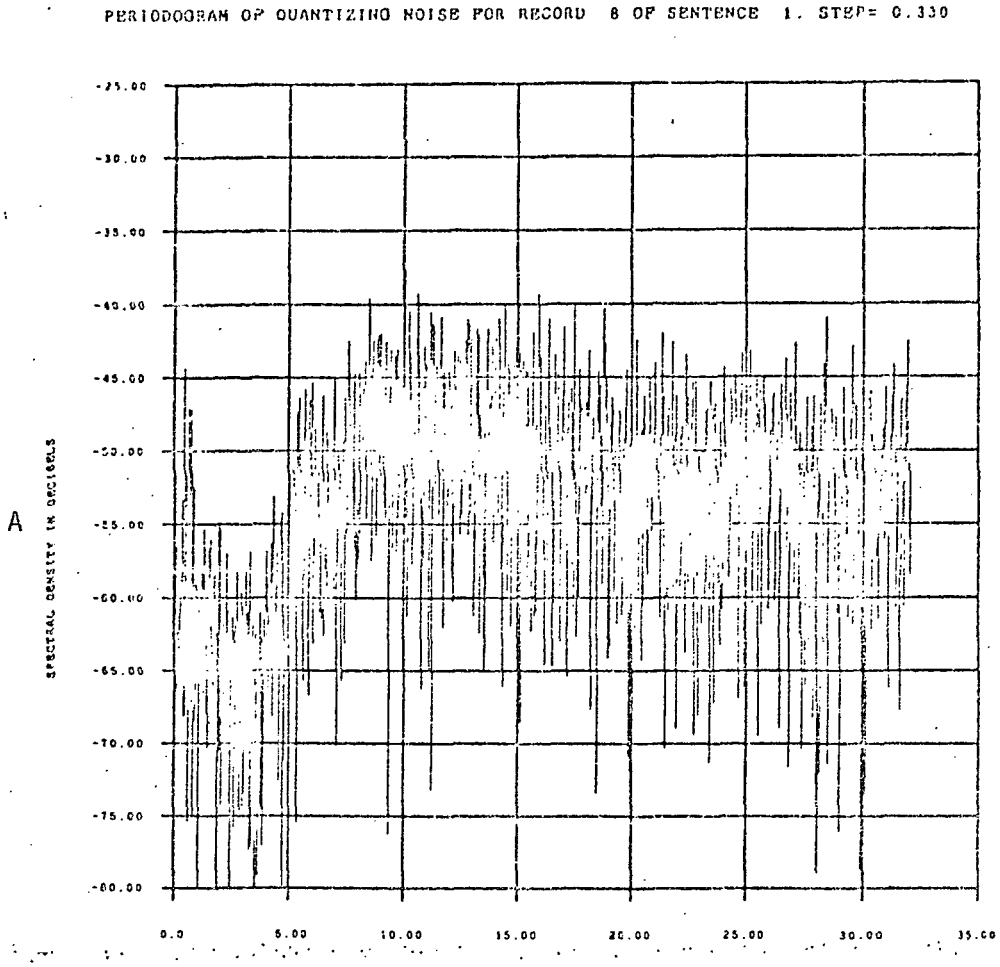



FIG. 5.7 NOISE SPECTRUM, RECORD 5, STEP=.43
A. STANDARD DM B. BLOCK-1, M=8



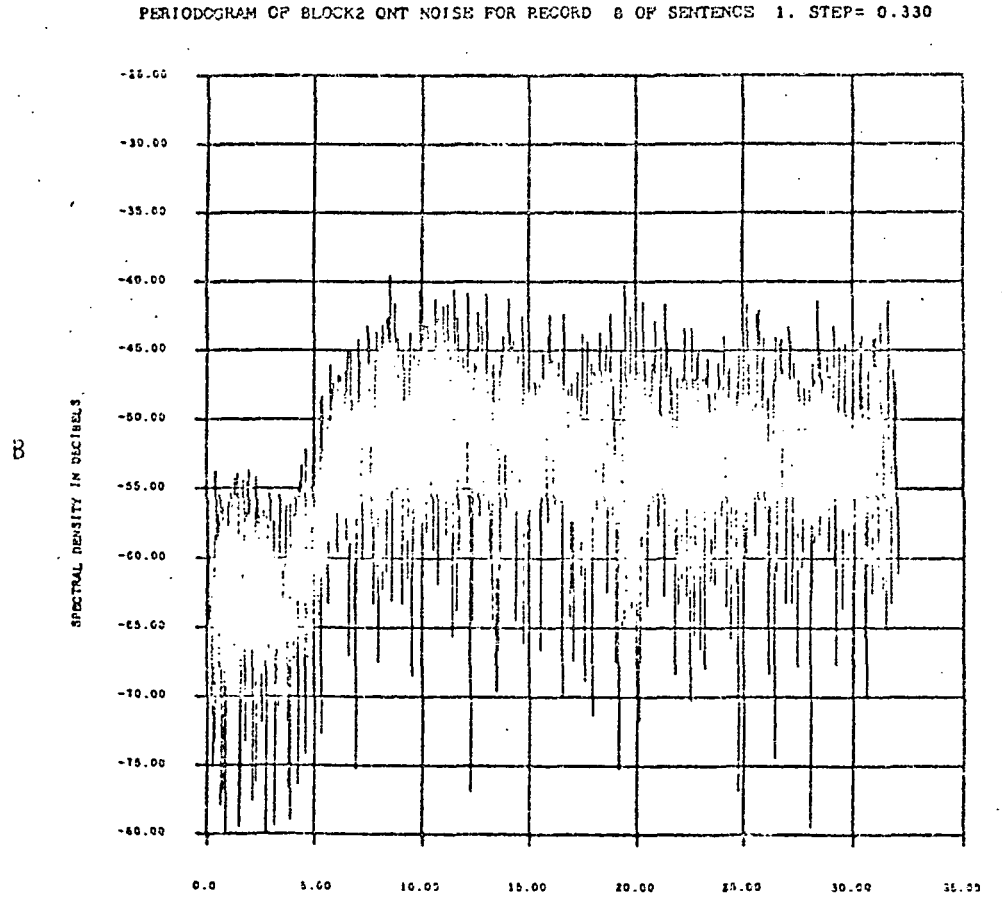
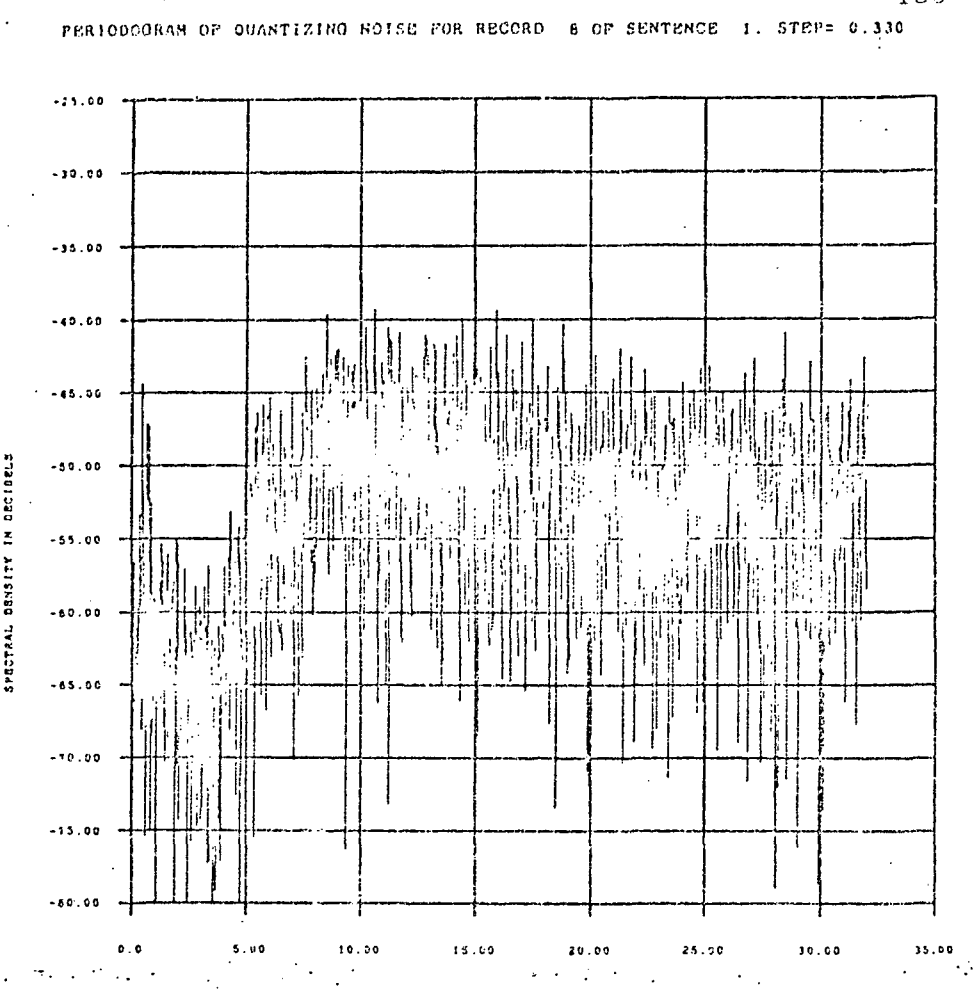
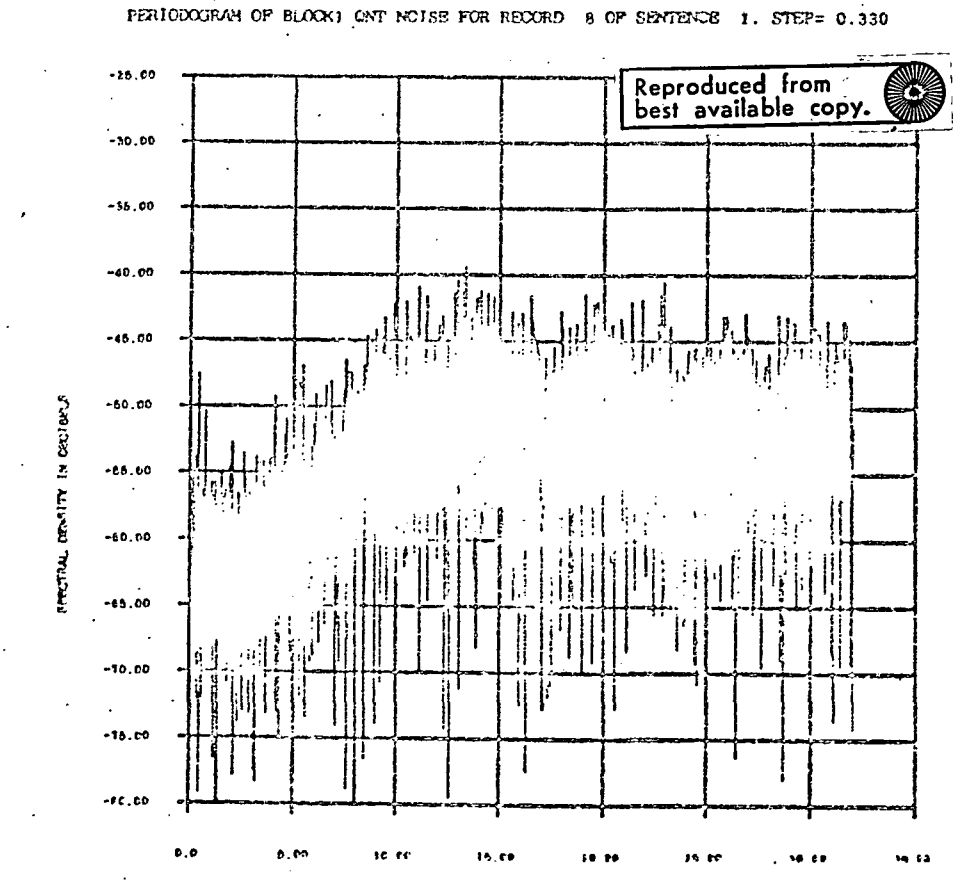




FIG. 5.8 NOISE SPECTRUM, RECORD 8, STEP=.33
A. BLOCK-1, M=24 B. BLOCK-2, M=23




A

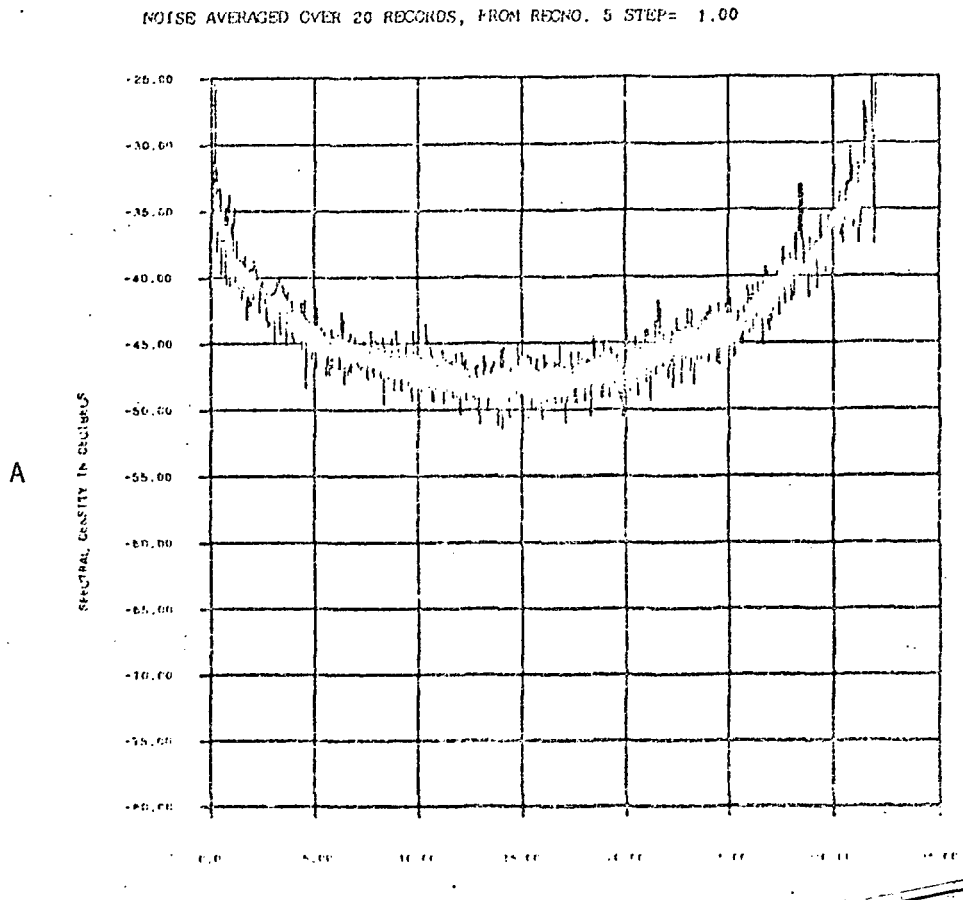





B

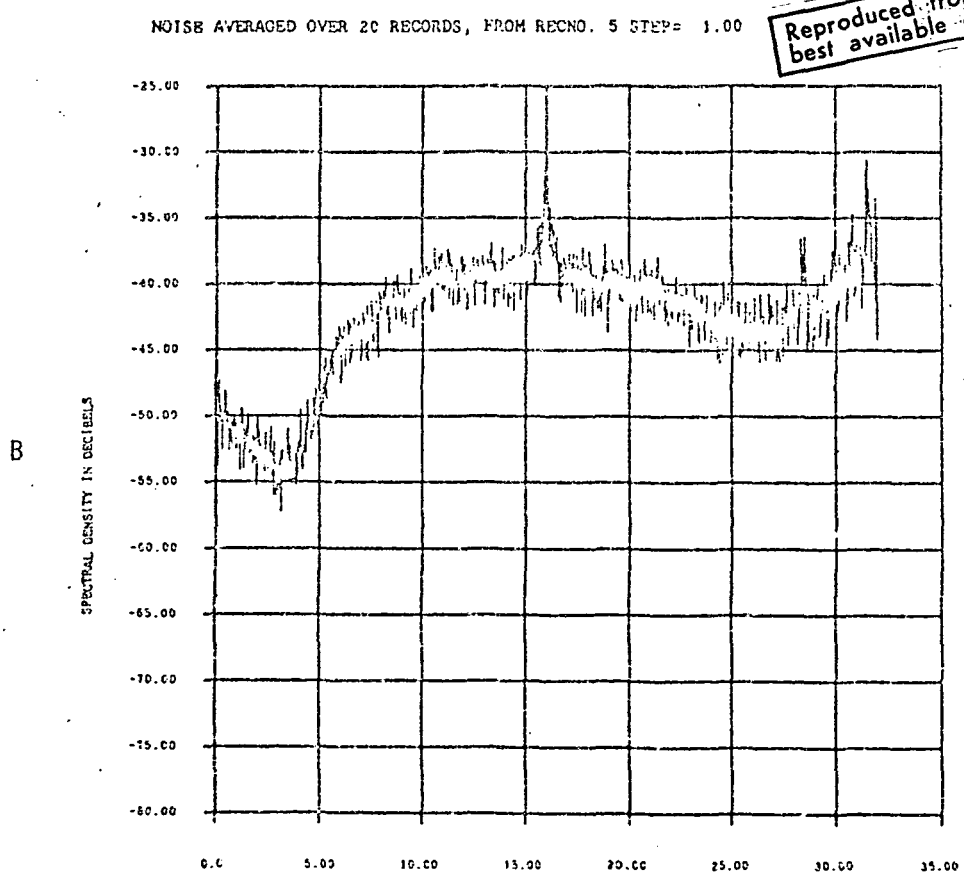
FIG. 5.9 NOISE SPECTRUM, RECORD 8, STEP=.33

A. BLOCK-1, M=24 B. BLOCK-1, RECURSIVE FILTER




A






B

FIG. 5.10 NOISE SPECTRUM AVERAGED OVER RECORDS 5 THRU 24.
A. STANDARD DM B. BLOCK-1, M=24

# REFERENCES


1. T. Fine, Properties of an Optimum Digital System and Applications, IEEE Trans. on Information Theory, Vol. IT-10 pp. 287-296, October, 1964.

2. H. Gish, Optimum Quantization of Random Sequences, Ph.D. Thesis, Harvard University, 1967.

3. A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw Hill, New York, 1965.

4. A. M. Yaglom, Theory of Stationary Random Functions, Prentice Hall, 1962.

5. C. C. Cutler, Transmission Systems Employing Quantization, U. S. Patent 2,927,962, March, 1960.

6. H. A. Spang and P. M. Schultheiss, Reduction of Quantizing Noise by Use of Feedback, IRE Transactions on Communication Systems, Vol. CS-10, No. 4, December, 1962, pp. 373-380.

7. E. G. Kimme and F. F. Kuo, Synthesis of Optimal Filters for a Feedback Quantization System, IEEE Transactions on Circuit Theory, Vol. 10, No. 3, September, 1963, pp. 405-413.

8. R. C. Brainard, Subjective Evaluation of PCM Noise-Feedback Coder for Television, Proceedings of the IEEE, Vol. 55; No. 3, March, 1967, pp. 346-353.

9. R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, New York, 1958.

10. Richard Bellman, Introduction to Matrix Analysis, McGraw Hill, New York, 1960, Appendix D, Moments and Quadratic Forms.

11. John M. Wozeneraft and Irwin Mark Jacobs, Principles of Communication Engineering, Wiley, 1967.

12. John E. Abate, Linear and Adaptive Delta Modulation, Proceedings of the IEEE, Vol. 55, No. 3, March, 1967, pp. 298-308.

13. S. J. Brolin and J. M. Brown, Companded Delta Modulation for Telephony, IEEE Trans. on Comm. Systems, Vol. COM-16, No. 1, February, 1968, pp. 157-162.

14. S. J. Brolin, Improving the Performance of Delta Modulators by Studying Their Performance for Realistic Nonrandom Inputs, Ph.D. Thesis, New York University, 1969.

# Chapter VI

## 6.1 Details of the Simulation

The complete computer simulation of a source encoding-decoding system is displayed in block diagram form in Figure 6.1. It is the purpose of this first section to trace through the process in greater detail, but still at the functional block level. Each of the blocks within the "IBM 360" part of the chain represents a separate subprogram, which required considerable attention to achieve efficiency and flexibility. However, some of these, for example the DRS-to-decimal conversion, and the tape writing routine, are purely programming exercises which have no theoretical bearing on the simulation per-se. Others, such as the sampling rate multiplication unit, posed both theoretical and programming problems. Only the theoretical questions will be taken up in detail, and this is done in subsequent sections covering sampling rate multiplication and division, the associated digital filtering, spectral analysis, and the encoder simulator parameters.

The actual programming is omitted, partly because there are enough differences between one computer and another to render incompatible programs written in a common source language (say F TRAN IV, as was the case here). Furthermore, much of the programming was concerned with interfacing between the IBM 360 and DRS, and would therefore not be of general interest. Enough theoretical information is given, nevertheless, to allow one to develop similar programs for his own

computer; and with the proper hardware available, one could reproduce, effectively, the present experimental setup.

The simulation process begins with sampling the analog source material at the 8 kHz Nyquist rate using the DRS (Digital Recording System), which is a machine built expressly to sample, quantize, and digitally record sound, with reproduction fidelity surpassing that of present day analog recorders. The source material is first sharply bandlimited to 4 kHz by an analog low-pass filter. Then it is sampled at 8 kHz, and quantized. A range of ±10 volts is allowed, with $2^{16}$ quantization levels available. The sample magnitude range 0 to 1 v. is linearly quantized into $2^{14}$ levels, and the range 1 v. to 10 v. is also given $2^{14}$ evenly spaced representative levels. This uses up 15 bits, and adding a sign bit makes 16 for the complete sample description. A sample magnitude exceeding 10 volts (overload) is assigned the appropriate largest code. Thus, in effect each sample is given a two-piece linear (dogleg) compression, followed by 16 bit uniform quantization. The 16 bits are written out in special DRS code as two consecutive 8 bit bytes on a 9 track magnetic computer tape. DRS also blocks the tape, and inserts inter-record gaps to make it compatible with the IBM 360.

The first step, upon reading the tape with the 360, is to convert the DRS binary code into floating point sample values. The final internal machine representation is a single

precision, FORTRAN IV variable, giving at least 7 decimal digits of precision. Therefore, the DRS quantization is certainly controlling, since the decimal representation has a 25 bit mantissa, including sign. The sequence of decimal sample values is run through the sampling rate multiplication program to expand 8-fold into a 64 kHz sequence, which is then saved in disk memory. A typical sentence has 25,000 Nyquist samples (occupying about 3-1/8 real time seconds), which increases to 125,000 samples at the 64 kHz rate. The reason for intermediate disk storage is that tape handling, and especially the sampling rate multiplication operation, are quite expensive. Repeated use of the 64 kHz samples of a sentence can be made more economically with fast disk access, and the costly rate multiplication need be done only once per sentence. The multiple outputs from the disk memory symbol are to indicate several different stored sentences, which may be easily selected for the simulation run.

The encoder simulator is fed the 64 kHz samples directly from the disk memory. For operation at 32 kHz, the encoder uses every second value, and to simulate 16 kHz sampling, it uses every fourth value in the data. The encoder program generates two output sequences, the reconstruction and the error. The error, or any other sequence for that matter, is optionally frequency analyzed. As indicated, graphical display is used. This is about the only way to sensibly assimilate the mass of information generated by the spectrum analysis.

To continue along the path which leads to analog audible output, the reconstruction sequence, (which is

generated by the local decoder within the encoder) is then divided down to an 8 kHz sequence. Next, these values must be converted back to DRS code and placed on an output tape. In this conversion, the reconstruction sample is mapped to the nearest 16 bit DRS code, which introduces on the average a half bit of roundoff error. Production of the blocked output tape in DRS format completes the IBM 360 part of the operation. The output tape is placed on the DRS tape drive for unblocking and analog conversion. The latter is done by a digital-to-analog (D/A) converter followed by an analog low-pass filter.

Essentially, the DRS acts as an interface between continuous time, analog input and output, and the real valued, discrete time sequences handled by the digital computer. It was designed to be transparent. Sixteen bit quantization, with companding, should give a signal to noise ratio due solely to DRS in excess of 90 dB. However, because of imperfect alignment between the A/D and D/A converters within DRS, the 1/2 bit roundoff error, and internal thermal noise, the effective quantization is closer to 13 bits. Nevertheless, this is sufficient to make the DRS generated quantization noise negligible. A back to back test, including both ends of DRS, the sampling rate multiplication and division, but by-passing the encoding block (ie, the companding with a short circuit around the encoder) produced an output with no audible background noise, and no discernible distortion of the speech.

## 6.2 Sampling Rate Multiplication or Division

The Block-N DM systems were simulated with encoding rates of 32 and 64 kHz, but the original analog speech material could only be physically sampled at 8 kHz because of hardware limitations. Consequently an interpolation operation was performed on the 8 kHz data in order to recover the required interstitial sample values. This amounts to reconstituting the original waveform by an expansion with an appropriate bandlimited interpolation function (sampling theorem), and then mathematically sampling that at any desired frequency. When the rates are integrally related, and a time undreated interpolation function is (necessarily) accepted, the process is equivalent to convolving a sequence derived from the 8 kHz data with a finite sequence related to a causal lowpass digital filter.

This may conveniently be described in the frequency domain. Consider a cascade of two samplers, schematized by the two momentary closing switches in Figure 6.2a, where one operates at 8 kHz and the other at 64 kHz. It should be clear that their order is unimportant, except that both switches are synchronized to be closed simultaneously every eight closures of the faster. The effect of the first sampler, operating at 8 kHz, is to generate a sequence whose spectrum is the spectrum of the source repeated with period 8 kHz. This is shown in figure 6.2c. The second sampler

takes this periodic spectrum and repeats it again, with the result of placing the sampling frequency between the eighth and ninth sections, as shown in figure 6.2d.

In order to convert the artificially derived 64 kHz sequence at the output of the second sampler into one which represents samples of the source, it is merely necessary to suppress the energy beyond 4 kHz in that sequence by a lowpass digital filter. Therefore, interpolation is equivalent to applying a lowpass filter to a 64 kHz sequence that is obtained by inserting seven zeroes between consecutive 8 kHz samples. This is suggested in figure 6.2e, where the unwanted "sidebands" have been reduced by the filter.

The effect of imperfect filtering, ie, nonzero sidebands, is to cause the interpolated sequence to be not strictly lowpass. However, if the filtering is adequate, the power in the source sequence above the signal band can be reduced to negligible levels. Consider the source spectra in figure 5.4, which are real examples of figure 6.2e. The filter, to be discussed in Section 6.3, has minimum out-of-band suppression of 80 dB. This puts the dominant in-band component at least 80 dB above its counterpart in the first sideband, making the spurious signal energy well below the expected quantizing noise levels. The point is that additional quantizing noise resulting

from encoding with some signal energy out of the nominal source band is assumed to be negligible with the present filtering.

The inverse to the sampling rate multiplication problem exists when the reconstruction sequence is ultimately to be converted to the lowpass, continuous time analog. Again, the problem arises because of hardware limitations: the device which translates the numerical sequence values into physical voltages can only operate at 8 kHz. The requirement, therefore, is to derive an appropriate 8 kHz numerical sequence from the higher rate, say 64 kHz, reconstruction sequence. And as in the previous problem, the solution involves a lowpass digital filtering operation.

If the 64 kHz numerical sequence were directly converted to the electrical equivalent at that higher rate, using an impulse modulator for example, the physical power spectrum is the same as the mathematical spectrum of the sequence (Section 1.2). In that case, decoding is completed by passing the 64 kHz impulse train through an analog lowpass filter, to remove the energy above the signal band, as well as to eliminate all the higher order segments of the periodic power spectrum.

On the other hand, if the reconstruction sequence were to have no energy beyond the signal band, then every eighth member forms an 8 kHz sequence, which may be converted to an 8 kHz impulse train, and analog lowpass filtered to obtain the continuous time output. The reconstruction sequence, however, contains quantizing noise

above the signal band. In that case an 8 kHz sequence trivially derived by retaining every eighth member contains the signal, plus the power sum of all the quantizing noise from 0 to the folding frequency, 32 kHz. This is because the 8 kHz sequence is actually a slow speed "sampling" of the relatively broadband 64 kHz sequence, which results in total aliasing whereby all spectral components above 4 kHz are brought back into the signal band. Therefore, if there is no such out-of-band noise originally, there will be no aliasing, and the 8 kHz sequence is then equivalent to the 64 kHz sequence.

Accordingly, sampling rate division is accomplished by first smoothing the reconstruction sequence with a low-pass digital filter, and then selecting every eighth member. Any out-of-band noise not removed will be aliased, and appear mixed with the signal in both the 8 kHz sequence and the continuous analog. Because the action of block-N weighted noise encoding results in decreased in-band noise at the expense of greatly increased out-of-band noise, the quality of this digital filtering is particularly important.

Continuing with the example of a rate ratio of 8, assume the out-of-band noise density is uniformly 15 dB above the in-band noise density. If the digital filter had only 20 dB stop band rejection, then the aliased noise would have the relative power spectral density#

---

# This highly simplified example overlooks pass-band-stop band transition width, etc., in assuming a rectangular filter. It is intended only to convey the essence of the argument.

$$15 - 20 + 10 \log 7 = 3.45 \text{ dB},$$

which raises the total in-band noise power at the filter output by approximately 5.1 dB. In order to maintain the noise degradation to less than .5 dB increase, the minimum stop band rejection must be 32.5 dB. This is easily met by the filter which was used, although the nonzero transition width would account for more than just .5 dB degradation, because of aliasing from the first sideband.

## 6.3 Digital Lowpass Filter and Window Function

An important component incorporated into several of the simulation subprograms is a nonrecursive digital lowpass filter. This filter serves first as an interpolator to expand the given 8 kHz source samples into a band-limited 64 kHz sequence of source samples, and then to remove the out-of-band quantizing noise from the reconstruction sequence to permit it to be collapsed back to an 8 kHz output sequence.

The filter was designed by the so-called modified Fourier series approach [15], wherein a nonrecursive filter is obtained by expanding the desired transmission function into a Fourier series, and then modifying the indicated tap weights by a window function to reduce the truncation error caused by retaining a finite number of terms. The resulting filter may require many more delay elements than a recursive type in order to meet the transmission requirements, but it is easier to design and analyze, and possesses better phase response.

Once having decided on the order of the filter, and the window function, the determination of the tap weights is straightforward. With x the input, and y the output process, the unmodified filter in the time domain is given by

$$y(j\tau) = \sum_{n=-D}^{D} a_n x(j\tau - n\tau), \qquad (6.3.1)$$

If these coefficients were used directly, the resulting filter transfer function would exhibit the familiar Gibbs oscillations, causing the sidelobes to be unduly large. This behavior is corrected by multiplying the $a_n$ by a window, which is a real, even function of n, whose transform is highly concentrated about zero frequency.

The window function used here is one advanced by Kaiser [16] as being both easy to use and nearly ideal in its shape. Actually, it is a one parameter family of functions

$$f(t,\alpha) = \begin{cases} \dfrac{\alpha I_o\left[\alpha\sqrt{1-t^2}\right]}{2 \sinh \alpha}, & |t| \leq 1 \\ 0, & |t| > 1 \end{cases} \qquad (6.3.6)$$

where $I_o$ is the zero order, modified Bessel function (of the first kind), and $\alpha$ is the parameter, which adjusts the tradeoff between central lobe width and sidelobe amplitudes in the spectral function. The above continuous time function is adapted to the present discrete filter by scaling its domain to $-D\tau \leq t \leq D\tau$, and then sampling with interval $\tau$, to obtain the coefficients

$$w_n = \frac{\alpha I_o\left[\alpha\sqrt{1-\left(\frac{n}{D}\right)^2}\right]}{2 \sinh \alpha}, \quad |n| \leq D, \qquad (6.3.7)$$

The time window (6.3.6) is shown in Figure 6.3a for $\alpha = 6$ and $\alpha = 8.5$, the latter used throughout. Figure 6.3b shows the corresponding spectral window, evaluated from the transform

$$F(\omega,\alpha) = \frac{\sinh\left[\alpha\sqrt{1 - \left(\frac{\omega}{\alpha}\right)^2}\right]}{\sinh(\alpha)\sqrt{1 - \left(\frac{\omega}{\alpha}\right)^2}}, \qquad (6.3.8)$$

which reveals the excellent bandlimiting property of this function.

Actually, this so-called $I_o$ - sinh window is nearly ideal because it is a remarkably close approximation to the prolate spheroidal function $S_{00}(c,t)$, which is known to have ideal bandlimiting properties [17] (minimum time-bandwidth product, etc.).

The filter, with modified tap weights $a_n w_n$, has a transmission function which is the ideal characteristic (6.3.3) convolved* with the spectral window (6.3.8). It may be shown [16] that the first order sidebands are down more than 80 dB, and from Figure 6.3b the main lobe width

---

* More precisely, it is a periodic convolution with an aliased spectral window,

$$\sum_{j=-\infty}^{\infty} F(\omega - j\omega_s, \alpha)$$

but the periodicity may be safely ignored when $D > 20$, as noted in [9], op. cit.

of the spectral window, which relates to the sharpness of the filter's cutoff, is $(2)\cdot(1.44)\cdot\left(\frac{64}{D}\right)$ kHz. For the interpolation filter used in sampling rate multiplication, D was chosen to be 100, giving a main lobe width of 1.84 kHz. That filter cuts off, therefore, in about 1.5 kHz. The parameter value 8.5 is toward the high end of the normal range of $\alpha$, which is approximately 4 to 9, and this is going in the direction of better stop band loss at the sacrifice of sharper cutoff.

In the interest of computation speed, a relatively short filter, D = 24, was adopted for sampling rate division. While still giving a minimum of 80 dB stop-band loss, the 6 dB down point with $\omega_c$ = 4 kHz is $4 + (1.44)\cdot\left(\frac{64}{24}\right) = 7.84$ kHz, which is well into the high quantizing noise part of the reconstruction sequence spectrum. Because of the relatively slow cutoff of this short filter used for sampling rate division, there was significant aliasing of quantizing noise down from the first sidebands, those between 4 and 12 kHz. Of course, this has nothing to do with the spectrum analysis results in Chapter V, but it allows a certain amount of additional voice-band quantizing noise in the analog output which should not be attributed to the source encoding. Better filtering here would have yielded even better audio results.

band-limited pulses. The function of the data window is to shape the filter responses in order to obtain the best estimator for the particular job at hand.

The first step is to compute, via the FFT, the modified discrete Fourier transform (DFT)

$$A(n) = \frac{1}{\Lambda} \sum_{j=0}^{\Lambda-1} Q(j)V(j)e^{-2\pi ij\,\frac{n}{\Lambda}} \qquad (6.4.1)$$

of the noise record Q, for example, where V is the data window normalized such that

$$\sum_{j=0}^{\Lambda-1} V^2(j) = 1. \qquad (6.4.2)$$

The modified periodogram is simply $|A_n|^2$, which estimates the power spectrum density at discrete frequencies

$$\omega_n = \frac{2n\pi}{\Lambda\tau}\ , \quad n = 0,\ 1,\ \ldots,\ \frac{\Lambda}{2}\ . \qquad (6.4.3)$$

It can be shown that

$$E\{|A_n|^2\} = \int_{-\pi/\tau}^{\pi/\tau} \Omega_Q(\omega)P(\omega-\omega_n)d\omega, \qquad (6.4.4)$$

where $\Omega_Q$ is the true power spectral density, and

$$P(\omega) = \left| \sum_{j=0}^{\Lambda-1} V(j)e^{i\omega j} \right|^2 \qquad (6.4.5)$$

is the spectral window. This is the familiar interpretation, namely, that one has an unbiased estimate of the true spectrum convolved with a spectral window.

However, in this case (as opposed to the indirect method whereby the autocorrelation is sought first) the estimator may not be a consistent one, ie, the variance of the estimates may not be driven to zero with increasing $A$, even though the stochastic process is ergodic. Nevertheless, the periodogram is still useful here. Realizing that the sequences to be analyzed, especially the speech samples, are nonstationary, let alone ergodic, a compromise data record length is picked to be long enough to give satisfactory stability to the estimates, while not so long that the local statistics of the data would change greatly. A value which gave good results is $A = 4096 = 2^{12}$, which happens to (conveniently) equal the 64 kHz sampling rate logical record length used in the simulation. Also, it was necessary that $A$ be a power of 2 in order that available FFT routines could be used; the FFT depends on the number of points to be a highly composite number to achieve the greatest computational efficiency.

Notice from (6.4.5) that the spectral window is the square of the Fourier cosine transform of the (symmetrical) data window. This is because it was applied to the data directly, whereas in the direct method the lag window is applied to the autocorrelation, and therefore gives a spectral window which is just its Fourier cosine transform.

The $I_o$-sinh window, with $\alpha = 8.5$, is employed here also. Therefore, the spectral window $P(\omega)$ is as shown in Figure 6.3b. To calibrate the frequency scale, set $D = \frac{\Lambda}{2}$, and thereby determine that the main lobe half width (first zero) is the solution to

$$\left(\frac{\omega}{2\pi}\right)\left(\frac{4096}{2}\right)\left(\frac{1}{64000}\right) = 1.44, \qquad (6.4.6)$$

which gives 45 Hz. This is approximately 3 times the spacing between estimates, $\frac{1}{\Lambda T} = 15.6$ Hz, which gives reasonable resolution. More importantly, the sidelobes are down a minimum of 60 dB, and that practically eliminates the problem of leakthrough which is a major concern when strong spectral components are in the vicinity of a relatively low level band being measured.

For the particular job of spectrum analysis of the quantizing noise, and of the source sequences for that matter, this characteristic was crucial. In the quantizing noise spectra, for example, the noise power is sometimes 20 dB greater just outside the source band. In order to observe this, the sidelobes of the spectral window must be particularly well behaved.

As seen in Figure 5.10, which is the average of many individual, modified periodograms taken of contiguous records, the estimate variance appears to be significantly reduced. This is expected for stationary ensembles, and is the

motivation for the approach in [18]. The fact that this was also seen for the noise over an interval of 1.38 seconds suggests that the noise is fairly (wide sense) stationary, for both standard DM and Block-1 DM. At least, this appears to hold when there is negligible overload.

## 6.5 Encoder simulation and parameters.

The encoder decision rule is easily programmed, following either the development in chapter 3 for the Block-1 types, or the methods given in chapter 4 for block lengths greater than one. In fact, with the stored program capability and floating point arithmetic of a digital computer, the simulation can be much less complicated than an actual hardware implementation, which uses threshold detectors and binary logic.

For instance, the simple comparison scheme with an ordered list of reconstruction candidates, described in section 4.1, is almost trivial to code into a source language such as FORTRAN. While this would be a completely valid simulation, and indeed the way to do the encoding if a programmed computer were to be the actual device, it would relate very little if at all to problems of hardware encoder design. Therefore, it was decided to simulate as much as practical, the detailed processes of the hardware encoder.

The Block-1 strategy can be programmed in just a few lines of FORTRAN IV, including the computation of $Y_n$. The following statements could form the nucleus of a

Block-1 DM sub-program. Identification of any undefined variable names will be obvious.

```
CHANGE = -DELTA
IF(SAMPLE + PSIO .GT. RECON) CHANGE = DELTA
RECON = RECON + CHANGE
ERROR = SAMPLE - RECON
```

The only part which requires some thought is the computing of $\gamma_0$, which is basically a numerical convolution. The usual technique is to use a circular storage array, herein named BUFFER, with which to store the past error values. The array is made circular by computing indices for it with modulo M arithmetic, where M is the number of past errors included in the $\gamma_0$ sum. The integer variable IPOINT keeps track of the current starting location within the array. The first step in each cycle is to update it in order to store the new error value, which over-writes the oldest one in the storage.

```
IPOINT = MOD(IPOINT, M) + 1
BUFFER(IPOINT) = ERROR
```

Finally, the new value of $\gamma_0$ is obtained by summing backwards in index over the past errors stored in BUFFER, weighted by the coefficients in the array B, Notice the computation of the index to locate each of the past errors, in order of increasing age.

```
      PSIO = 0.
      DO 100 J = 1, M
      INDEX = MOD(IPOINT + M - J, M) + 1
100   PSIO = PSIO + BUFFER(INDEX) * B(J)
```

At this point, PSIO has been updated to be ready for the next decision, and program control would now transfer back to bring in the next source sample. Standard DM is simulated simply by excluding the above code, except for the initialization PSIO = 0.

The Block-3 sub-program followed the stream mode encoder concept exactly. Three subsections were written which simulate the first digit units. The commutating switch is incorporated into the program by transferring to the next subsection, depending on which one was just exited. After a given first digit unit simulator makes its decision, control is passed to a common section of the subprogram which updates RECON, subtracts it from SAMPLE to generate the new ERROR, and then computes the required bias parameters. Control then passes to the next first digit unit statements in proper sequence.

This arrangement allows the simulation to take advantage of some of the properties of the stream mode design. For example, Block-2 is simulated by having program control pass to the N=2 first digit section after the block-1 decision is made, instead of to the N=2 first digit subsection. Also, the same approximations

can be made, whereby less consequential boundaries are omitted from consideration in the higher order first digit subsections.

This part concludes with a few remarks on the weighting function, which along with M and N is one of the major choices to be made in the encoder design. Throughout, the basic unity in-band rectangular noise weighting has been considered. The effective noise weighting, as discussed in section 2.2, is this ostensible characteristic convolved with the periodic window, eq. 2.2.15, which is implied whenever the B matrix entries are computed directly as the Fourier coefficients of the ostensible weighting.* This is illustrated in figure 6.4, which includes for comparison the effective weighting function obtained if the triangular lag window (2.2.2) had not been used in the derivation of B. The block length, L, is 25 in this example.

The question arises whether the overshoot characteristic is desireable, or inferior in performance to the non-negative effective weighting. In general, the optimum effective weighting, which is constrained to be a L - order polynomial in $\cos(\omega/\omega_s)$, is not known. Another interesting question involves specialized weighting functions, which

---

* [illegible] of the [illegible] solution. The ostensible [illegible] is convolved with the Fejer kernel, which gives bandlimited approximation without overshoot. In this case it maintains the effective weighting positive.

penalize the noise selectively within the source band. It can be seen that when the source band is a small fraction of the reconstruction band, ie a large sampling rate/ Nyquist rate ratio, it will not be possible to achieve an effective weighting function which approximates fine structure in the ostensible function, with small or moderate L values.

Optimization of the effective weighting function is an interesting problem for further investigation, which is closely tied to the question of total block length and the encoding dimensionality, N. The answers, in large measure, will be determined by future technology. It may be, for example, that M=100 is not unreasonable, in which case bandlimited approximation of the weighting function is not a major concern.

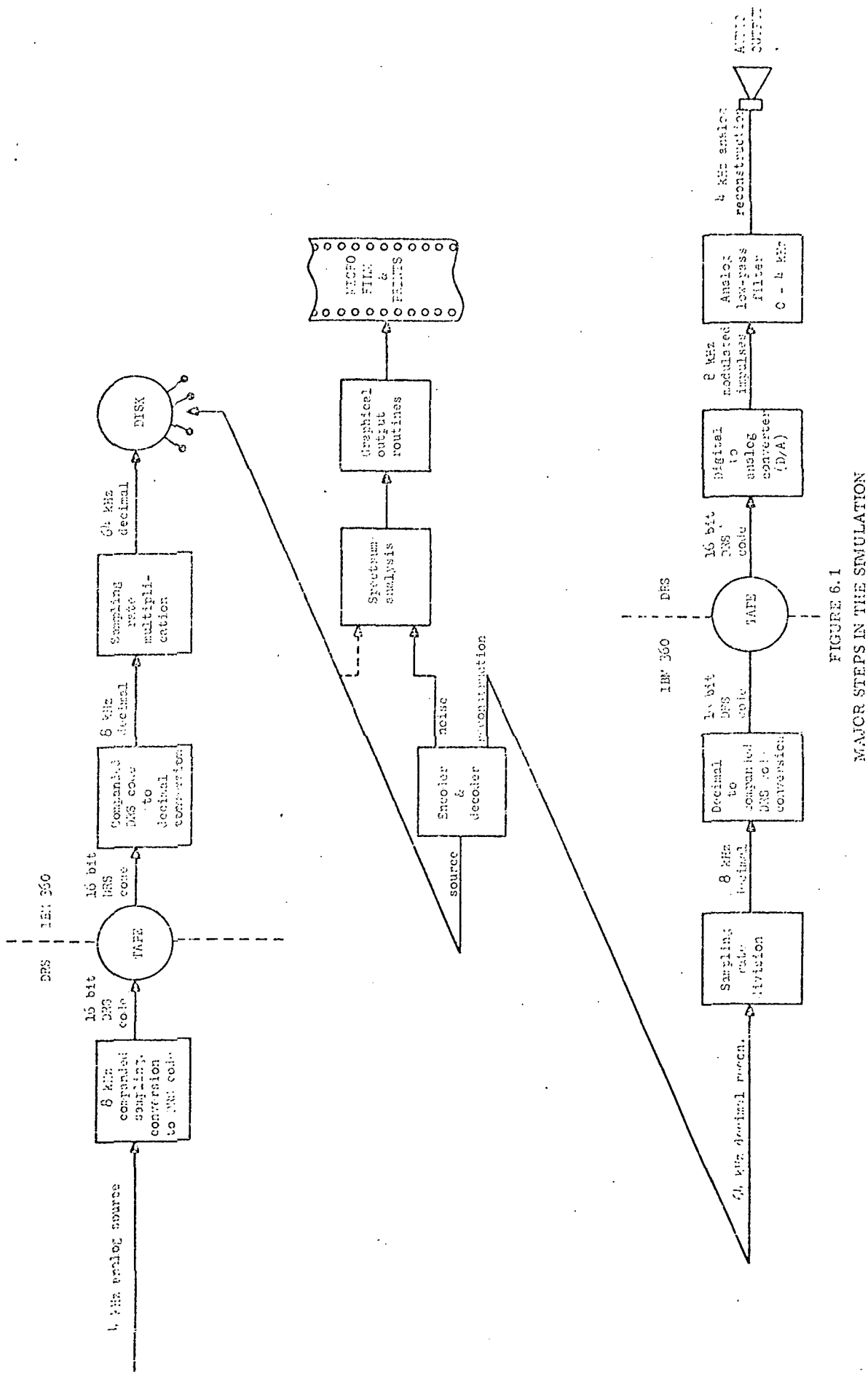
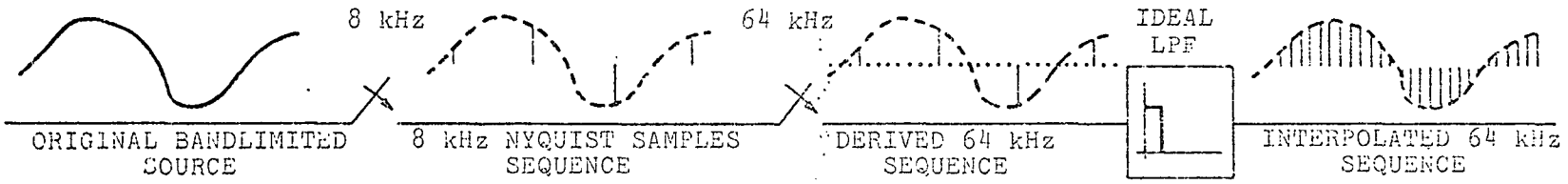
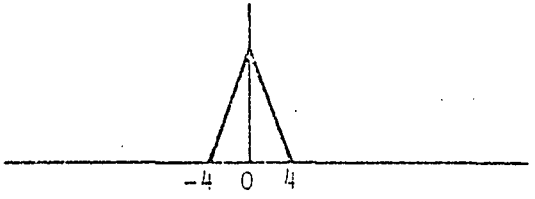


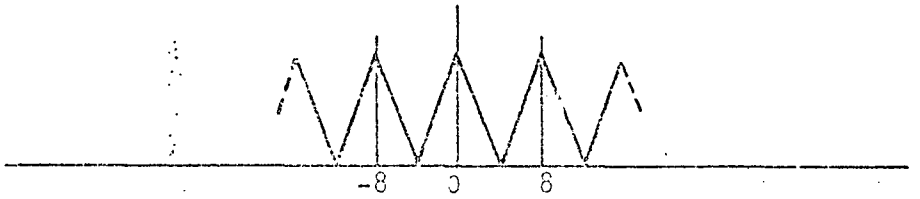

FIGURE 6.1
MAJOR STEPS IN THE SIMULATION



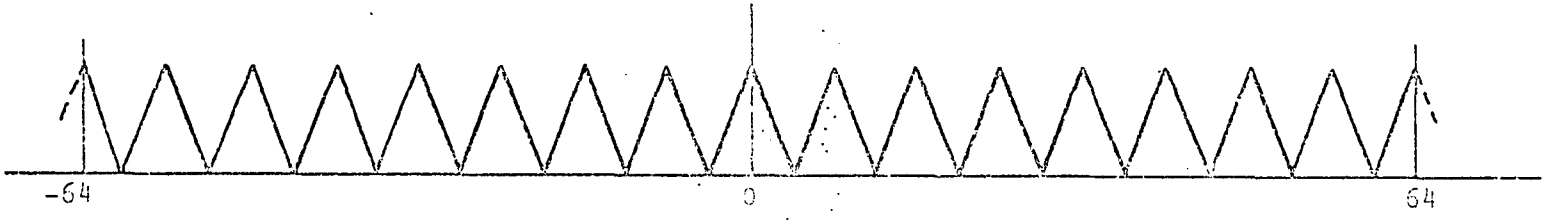

A. STEPS IN THE TIME DOMAIN



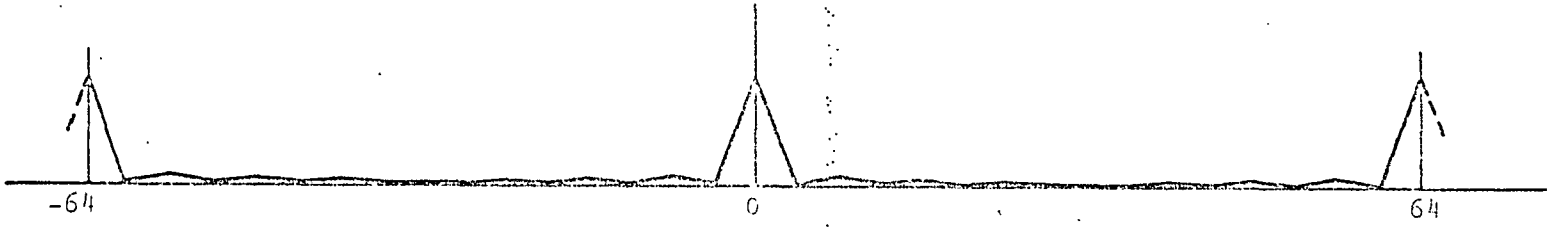

B. CONTINUOUS SOURCE SPECTRUM




C. NYQUIST SAMPLE SPECTRUM

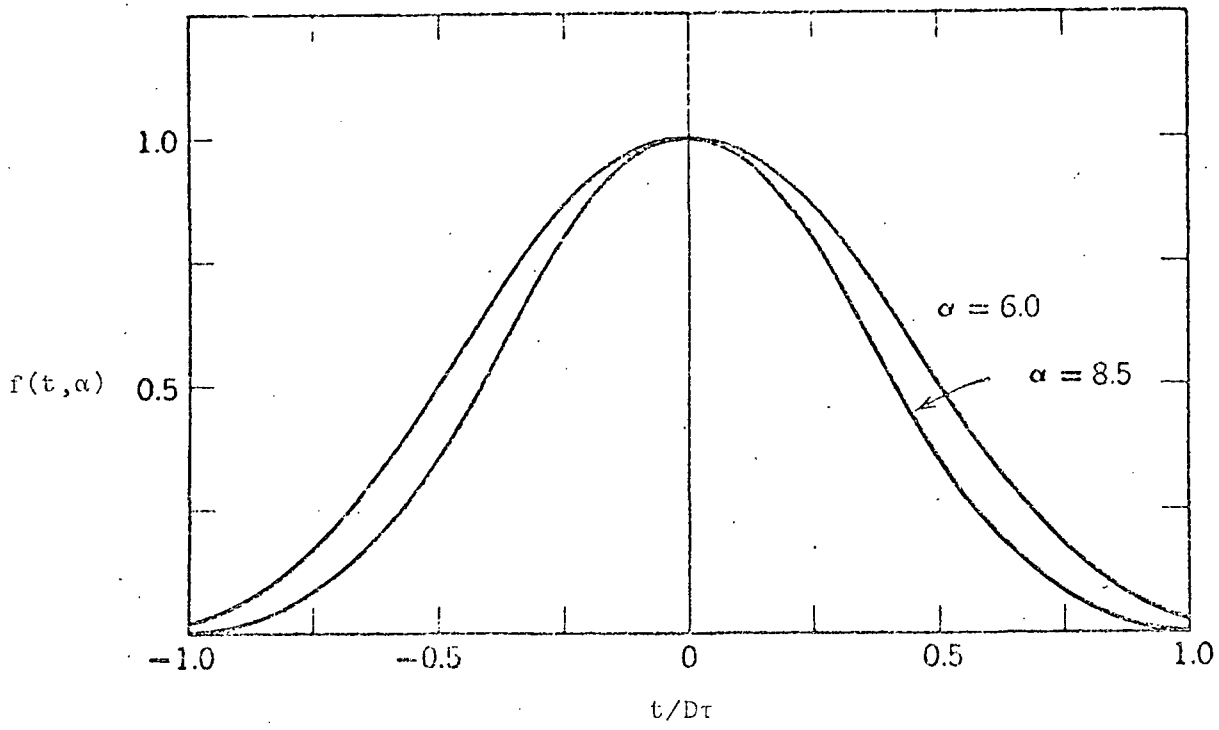



D. SPECTRUM OF DERIVED 64 kHz SEQUENCE

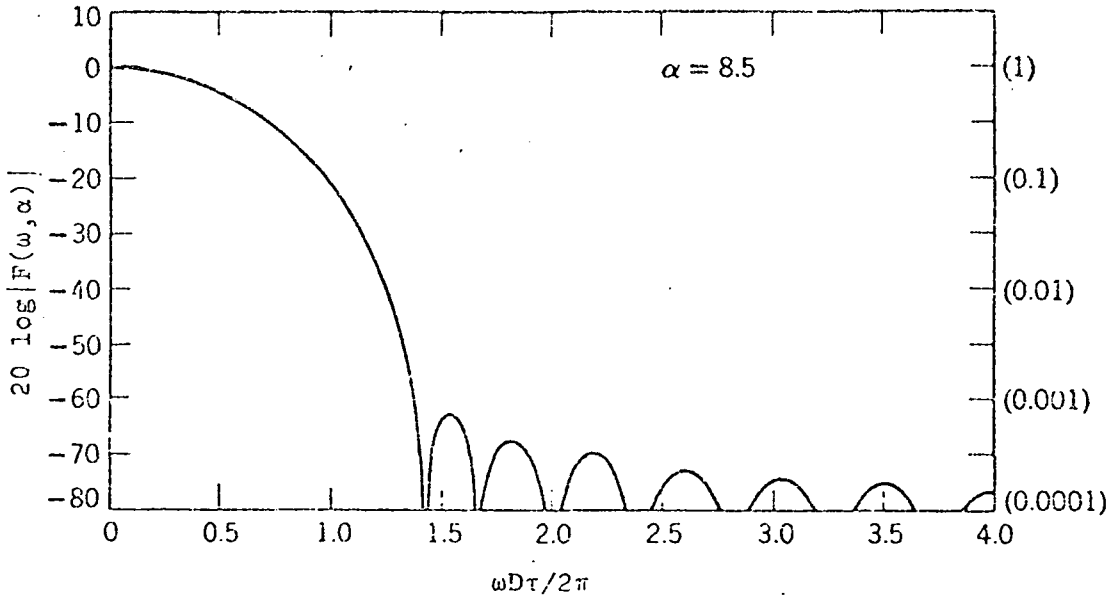



E. INTERPOLATED SEQUENCE SPECTRUM AFTER NON-IDEAL FILTERING

FIGURE 6.2 SAMPLING RATE MULTIPLICATION




A. TIME WINDOW




B. SPECTRAL WINDOW

FIGURE 6.3
$I_0$-SINH WINDOW PAIR

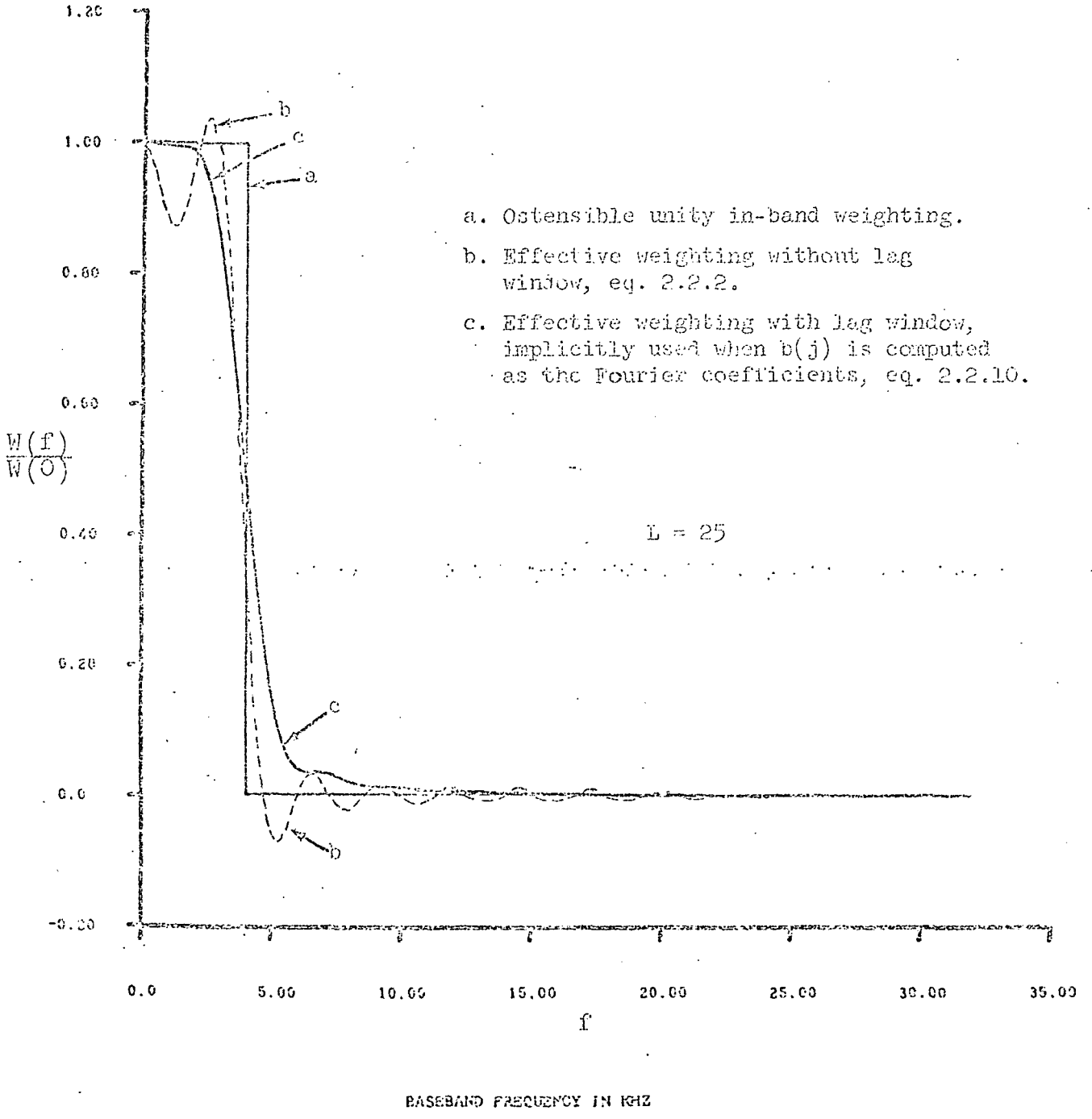



FIGURE 6.4

NOISE FREQUENCY WEIGHTING FUNCTIONS

15. F. F. Kuo and J. F. Kaiser, editors, System Analysis by Digital Computer, Wiley, 1966, Chapter 7, Digital Filters by J. F. Kaiser.

16. J. F. Kaiser, A Family of Window Functions Having Nearly Ideal Properties, unpublished.

17. D. Slepian and H. O. Pollak, Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty - I and II, B.S.T.J. Volume 40, No. 1, January 1961, pp. 43-84.

18. P. D. Welch, The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodogram, IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, No. 2, June 1967, pp. 70-73.

15. F. F. Kuo and J. F. Kaiser, editors, System Analysis by Digital Computer, Wiley, 1966, Chapter 7, Digital Filters by J. F. Kaiser.

16. J. F. Kaiser, A Family of Window Functions Having Nearly Ideal Properties, unpublished.

17. D. Slepian and H. O. Pollak, Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty - I and II, B.S.T.J. Volume 40, No. 1, January 1961, pp. 43-84.

18. P. D. Welch, The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodogram, IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, No. 2, June 1967, pp. 70-73.

III. New Research

During the next year, research into the use of the Song adaptive DM will be extended to consider its response to video signals. In addition, techniques which will increase SNR will be tried. For example we shall insert a shaping filter before the limiter in the adaptive delta modulator to shape the spectrum of the error. The Nth-order DM will also be employed with the adaptive DM. In this regard we hope to time-share the same DM rather than use N modulators.

The video signal in these experiments will be obtained using a flying spot scanner which is available in our laboratory.

In addition we shall begin to consider the effect of channel noise on the response of a DM.

The Rosenbaum DM will be constructed and tested for voice signals. The shaping filter will be inserted in the linear and adaptive DM systems available. Voice tapes will be made to compare systems.

III. New Research

During the next year, research into the use of the Song adaptive DM will be extended to consider its response to video signals. In addition, techniques which will increase SNR will be tried. For example we shall insert a shaping filter before the limiter in the adaptive delta modulator to shape the spectrum of the error. The Nth-order DM will also be employed with the adaptive DM. In this regard we hope to time-share the same DM rather than use N modulators.

The video signal in these experiments will be obtained using a flying spot scanner which is available in our laboratory.

In addition we shall begin to consider the effect of channel noise on the response of a DM.

The Rosenbaum DM will be constructed and tested for voice signals. The shaping filter will be inserted in the linear and adaptive DM systems available. Voice tapes will be made to compare systems.

IV. Doctoral Dissertations

During the period of this grant PhD Dissertation having research partially supported by NGR 33-013-063, have been conpleted by:

1. A. S. Rosenbaum, "Source Encoding With a Frequency Weighted Error Criterion"
2. J. Garodnick, "Digital Processing in Communication Systems"

IV. Doctoral Dissertations

During the period of this grant PhD Dissertation having research partially supported by NGR 33-013-063, have been conpleted by:

1. A. S. Rosenbaum, "Source Encoding With a Frequency Weighted Error Criterion"
2. J. Garodnick, "Digital Processing in Communication Systems"

## V. Papers Published

1. "Robust Delta Modulation" Proceedings of the IEEE Fall Electronics Conference, October 1971 (with J. Garodnick and C. L. Song)
2. "A Variable Step-Size Delta Modulator", IEEE Transactions on Communication Technology, December 1971 (with C. L. Song and J. Garodnick.
3. "Source Encoding with a Frequency Weighted Noise Criterion" (with A. S. Rosenbaum), IEEE International Information Theory Symposium, February 1, 1972.

V. Papers Published

1. "Robust Delta Modulation" Proceedings of the IEEE Fall Electronics Conference, October 1971 (with J. Garodnick and C. L. Song)
2. "A Variable Step-Size Delta Modulator", IEEE Transactions on Communication Technology, December 1971 (with C. L. Song and J. Garodnick.
3. "Source Encoding with a Frequency Weighted Noise Criterion" (with A. S. Rosenbaum), IEEE International Information Theory Symposium, February 1, 1972.

THE CITY COLLEGE RESEARCH FOUNDATION

THE CITY COLLEGE of THE CITY UNIVERSITY of NEW YORK

Convent Avenue at 138th Street, New York, New York 10031

Telephone: (212) 281-0470; (212) 368-1444