APPLICATIONS OF SATELLITE TECHNOLOGY FOR REGIONAL ORGANIZATIONS
(Project ASTRO)

Technical Report

November 1976

PRINCIPAL INVESTIGATORS

Donald L. Schilling, Ph.D.
Professor of Electrical Engineering
The City College
The City University of New York
New York, N.Y. 10031

Stanley C. Wecker, Ph.D.
Associate Professor of Biology
The City College
The City University of New York
New York, N.Y. 10031

under
NASA GRANT NSG 7144

TECHNICAL MONITOR

Mr. Herb Ernst, Manager
Data Management
National Aeronautics and Space Administration
Washington, D.C. 20546

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U S DEPARTMENT OF COMMERCE
SPRINGFIELD, VA 22161

THE CITY COLLEGE OF
THE CITY UNIVERSITY of NEW YORK

(NASA-CR-149184) APPLICATIONS OF SATELLITE TECHNOLOGY FOR REGIONAL ORGANIZATIONS (PROJECT ASTRO) (City Coll. of the City Univ. of New York.) 225 p HC A10/MF A01 CSCL 17B G3/32

N77-11267

Unclas
54504

INTRODUCTION

This technical report completes one facet of Project ASTRO. In this report we show that signals encoded in a delta modulator format can undergo arithmetic processing without first being transformed into a PCM format. In addition, we show that PCM and DM signals can be converted to the other format with less than 0.5 dB degradation.

Since a signal can be encoded into a DM format using half the number of binary digits required for PCM conversion, the results presented here indicate a way of significantly reducing the memory requirements of a data base. The next facet of this study will be to actually encode the data into a DM format store the bits in the (DM) data base and use that reduced data in the DBMS.

ARITHMETIC PROCESSING AND DIGITAL CONVERSION

OF

ADAPTIVE DELTA MODULATION ENCODED SIGNALS

by

JOSEPH L. LoCICERO

iv

ARITHMETIC PROCESSING AND DIGITAL CONVERSION

OF

ADAPTIVE DELTA MODULATION ENCODED SIGNALS

by

JOSEPH L. LoCICERO

A dissertation submitted to the Graduate
Faculty in Engineering in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy, The City University of New York.

1976

ib

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

10/25/76
date

Chairman of Examining Committee

10/25/76
date

Executive Officer

Dr. J. Garodnick
Dean R. B. Marsten
Prof. S. J. Oh
Prof. H. Taub
Dr. S. B. Weinstein

Supervisory Committee

ABSTRACT

ARITHMETIC PROCESSING AND DIGITAL CONVERSION

OF

ADAPTIVE DELTA MODULATION ENCODED SIGNALS

by

Joseph L. LoCicero

Advisor: Professor Donald L. Schilling

This thesis can be divided into three distinct parts: direct arithmetic processing of adaptive delta modulation (ADM) encoded signals, conversion from ADM encoded signals to pulse code modulation (PCM) encoded signals and conversion from PCM to ADM encoded signals. In the first part, it is shown that signals which are ADM encoded can be arithmetically processed directly, without first decoding. Operating on the DM bit stream, and employing only standard digital hardware, the sum, difference and product can be obtained in PCM and ADM format.

These arithmetic processing systems are analyzed and simulated on a digital computer. Employing a four-term, non-recursive, averaging filter after the processors, we show that, for constant inputs, the signal-to-noise ratio (SNR) of the DM device is exactly the same as that of their

PCM counterparts. SNR curves are obtained for these processors when the inputs are single frequency tones and their performance is compared to similar PCM systems. At high bit rates, the performance of the DM adder is comparable to that of an equivalent, companded PCM system. At moderate and low bit rates, the SNR of the DM adder is at least 4 dB better than that of the PCM device. The DM multiplier, at any bit rate, has a SNR which is at least 5-10 dB higher than the SNR of a companded PCM system.

The conversion from ADM to PCM encoded signals essentially deals with the changing from a high "information" rate (ADM) to a lower one (PCM). The "information" rate is the frequency at which estimates of the encoded signal are available. This problem is solved by using a digital filter, which operates on the ADM step sizes and which is designed with ideal low pass filter characteristics, and then utilizing a low (PCM) frequency sampling gate. Since the ADM estimate, $\hat{x}(k)$, is a wideband signal, even if the input is bandlimited, we must eliminate the high frequency components before sampling at the PCM rate or else suffer the disastrous effect of aliasing. We show the effect of the digital filter structure on $\hat{x}(k)$ by recasting the system into a cascade arrangement and evaluating, in the frequency domain, the transfer function relating $\hat{x}(k)$ to the improved ADM estimate.

The ADM to PCM conversion system, with the non-recursive digital filter structure, is restricted to standard digital

hardware and its operation is evaluated via computer simulation. The performance of this converter is compared to the performance of other systems which will produce the same net result. The simplest system achieves ADM to PCM conversion by sampling $\hat{x}(k)$ at the PCM rate. A family of performance curves, obtained with a sinusoidal input, shows that there is an 8-10 dB improvement in SNR, over the simple system, when the non-recursive digital filter structure is employed in the ADM to PCM converter. We find that the SNR of the optimal converter, which uses ideal analog demodulation of $\hat{x}(k)$ before PCM sampling, is only 1-2 dB better than the SNR of the converter with the non-recursive digital filter structure.

When converting from PCM to ADM encoded signals, i.e., changing from a low to a high "information" rate, we must estimate the signal excursion at discrete points between the PCM samples and fit these values to a path through an "ADM signal estimate tree." A number of techniques, both dependent on the input signal statistics and independent of them, are developed to perform this conversion. A detailed statistical analysis is undertaken for one particular system which employs a very simple, all-digital method of converting from PCM to ADM format. This converter uses samples of a linear interpolation between PCM points as the input to an ADM. This system is simulated on a digital computer and SNR curves are generated to determine its performance. We find that for a low frequency tone, the performance of this converter is

almost as good as if an ADM encoded the original tone. However, for a high frequency tone, the performance is unacceptable.

We introduce a non-parametric technique to estimate the midpoint between PCM samples using four adjacent PCM points. The estimator weights the four adjacent PCM points as if they were impulses passing through an ideal low pass filter. A linear interpolation is formulated between the PCM points and the estimated midpoint. Samples of this interpolation are then used as the input to an ADM to achieve conversion. The SNR of the non-parametric converter, for both low and high frequency sinusoids, comes within 1 dB of the SNR for the optimal system, i.e., when the ADM encodes the original tone. This converter utilizes only the standard digital hardware in its realization and stresses a simple structure that can be fabricated on a single, large-scale, integrated circuit chip.

HIGHER ATTAINMENTS

If I do this, what further can I do?
Why, more than ever. Every task thou dost
Brings strength and capability to act.
He who doth climb the difficult mountains
Will the next day outstrip an idler man.
Dip thy young brain in wise men's deep discourse,
In books which, though they freeze thy wit awhile,
Will knit thee in the end with wisdom.

BARRY CORNWALL

From *Viaticum: Apt Words Fitly Spoken*, "28th April"
(T. and A. Constable, Edinburgh, Scotland, 1918)

ACKNOWLEDGEMENTS

The author would like to express his sincere gratitude to Professor Donald L. Schilling for his faithful guidance, poignant criticism and personal encouragement during the entire course of this research.

The author also wishes to thank Dr. Joseph Garodnick, for his diligent reading of the first draft and for his ever searching questions, and Dean Richard B. Marsten, for his continuous interest in the progress of this thesis and for his helpful editorial remarks.

In addition, the author would like to thank all his colleagues in room T 502F, for their never-ending discussion of any aspect of this research, particularly Mr. Donald Ucci, for his attentive ear to all of the author's "easy" questions.

Finally, the author wishes to thank his loved ones, for their unending patience, Mr. F. Alan Pastore, for his superb typing of the thesis, and Mr. Nathaniel Silber, for his assistance in reproduction of the final manuscript.

The research contained in this dissertation was partially supported by the National Aeronautics and Space Administration under grants NSG 7144, NSG 5013 and NSA 9-13940.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

TABLE OF CONTENTS

Page

Ch. 1 INTRODUCTION 1

Ch. 2 DIRECT ARITHMETIC PROCESSING OF ADM ENCODED SIGNALS 6

2.1 The Basic Digital DM 6
2.2 Direct DM Addition/Subtraction 9
2.3 Averaging Filter 15
2.4 Direct DM Multiplication 23
2.5 Hardware Complexity 34
2.6 SNR with Constant Inputs 37
2.7 Simulation Results with Elementary Signals. 43
2.8 Performance Evaluation 51
2.8.1 Fourier Series Representation of the DM Estimate 53
2.8.2 Output SNR 56
2.8.3 Performance Curves 58
2.9 Comparison with PCM Systems 63

Ch. 3 CONVERSION FROM ADM ENCODED SIGNALS TO PCM ENCODED SIGNALS USING DIGITAL FILTER TECHNIQUES 72

3.1 Basic ADM-PCM Conversion Philosophy 72
3.2 ADM Encoder Technique 75
3.3 Non-Recursive Digital LPF Technique 79
3.3.1 Realizing an Ideal Digital LPF . . . 88
3.3.2 Characteristics of the Ideal Digital LPF 91
3.3.3 Converter Simulation and Sinusoidal Response 99
3.3.4 Evaluation of Performance 100
3.4 Other Digital Conversion Techniques 104
3.5 Analog Demodulation Technique 106
3.6 Comparison of Conversion Systems 110
3.6.1 SNR Evaluation 110
3.6.2 Three Families of SNR Curves 112

Ch. 4 CONVERSION FROM PCM ENCODED SIGNALS TO ADM ENCODED SIGNALS 116

4.1 DM Signal Estimate Tree 117
4.1.1 Paths through the Signal Estimate Tree 117
4.1.2 Path Endpoints 118

TABLE OF CONTENTS (continued)

Page

4.2 Statistical PCM-ADM Converter 118
4.2.1 Choosing Tree Path Endpoints 122
4.2.2 Choosing the Most Likely Path . . . 123
4.3 Parametric PCM-ADM Converter 125
4.3.1 A Simple, All-Digital Technique . . 126
4.3.2 Optimization of Converter Performance 127
4.3.3 SNR Statistical Analysis 134
4.3.4 Converter Improvement via Wiener Linear Interpolation 144
4.3.5 Performance of the Improved Converters 153
4.3.6 Simulation of the Fundamental Converter 161
4.3.7 SNR Curves 166
4.4 Non-Parametric PCM-ADM Converter 170
4.4.1 PCM Estimation Technique 170
4.4.2 Ideal LPF Impulse Response Weighting 171
4.4.3 Converter with Midpoint Estimate . . 173
4.4.4 Simulation and Performance 174
4.5 Other PCM-ADM Converters 179
4.5.1 Digital Zero-Order Hold Circuit . . 180
4.5.2 Submultiple Sampling Technique . . . 183
4.5.3 Recent Developments 185

Ch. 5 CONCLUSIONS 187

App. 1 AMPLITUDE-FREQUENCY CHARACTERISTICS OF THE FOUR-TERM NON-RECURSIVE AVERAGING FILTER . . . 191

App. 2 THE PCM PRODUCT VIA ALTERNATE TECHNIQUES . . . 193

App. 3 FREQUENCY CHARACTERISTICS OF THE CASCADE ARRANGEMENT FILTER, $H_{CD}(f)$, USED IN ADM TO PCM CONVERSION 198

App. 4 THE EFFECT OF TIME DELAY ON THE SNR OF THE BASIC PCM TO ADM CONVERTER 203

REFERENCES 207

LIST OF TABLES

Table		Page
2.9-1	Comparison of DM and PCM Arithmetic Processors	67
3.3-1	Coefficients for the Non-Recursive Digital LPF	92
4.3-1	Performance of Parametric PCM to ADM Converters	157

LIST OF ILLUSTRATIONS

Figure		Page
2.1-1	Basic Digital Delta Modulator	8
2.2-1	Direct Sum of DM Encoded Signals	11
2.2-2	DM Adder for Linear Mode	13
2.2-3	DM Adder for Song Audio Mode	14
2.3-1	Steady State Estimate Signal for the Song Audio Mode DM to a Constant Input	17
2.3-2	Steady State Direct Sum for the Song Audio Mode DM with Constant Inputs	19
2.3-3	Four-Term Averaging Filter Realization	21
2.3-4	Amplitude-Frequency Characteristics of the Four-Term Non-Recursive Averaging Filter	22
2.4-1	Direct Product of DM Encoded Signals	25
2.4-2	DM Multiplier for Linear Mode	27
2.4-3	Example when the Step Size Relationship is Invalid	29
2.4-4	Steady State Direct Product for Song Audio Mode DM with Constant Inputs	32
2.7-1	DM Sum of Step and Pulse	45
2.7-2	DM Sum of a Step and a Sinusoid	46
2.7-3	DM Sum of Two Sinusoids	47
2.7-4	DM Product of a Step and a Pulse	48
2.7-5	DM Product of a Step and a Sinusoid	49
2.7-6	DM Product of Two Sinusoids	50
2.7-7	DM Product of a Step and a Pulse without Four-Term Averaging Filter	52
2.7-8	Song Audio Mode DM Response to a Step	60
2.8-1	Song Audio Mode ADM Response to a Constant Input	61
2.8-2	SNR for DM Direct Sum	62
2.8-3	SNR for DM Direct Product	64
3.1-1	ADM Estimate Converter	76
3.3-1	Improved ADM Estimate Converter	81
3.3-2	ADM to PCM Converter with Non-Recursive Filter	84
3.3-3	Realization of the Product $g(j)S_x(k - j)$	86
3.3-4	Modified Realization of ADM to PCM Converter with Non-Recursive Filter	87
3.3-5	Unit Step Response for an ILPF with Zero Time Delay	89
3.3-6	Cascade Arrangement of the ADM to PCM Converter	93
3.3-7	Amplitude and Phase Characteristics of Converter Filter where R = 8, Q = 10	96
3.3-8	Amplitude Characteristics of Converter Filter where (a) R = 6, Q = 8, (b) R = 4, Q = 6	97

LIST OF ILLUSTRATIONS (continued)

Figure		Page
3.3-9	ADM Estimate: Before and After Non-Recursive Filter	101
3.5-1	Analog Demodulation Converter	107
3.6-1	Performance Curves for ADM to PCM Converters	113
4.1-1	Possible Paths Through an "ADM Signal Estimate Tree"	118
4.3-1	A Basic PCM to ADM Converter	127
4.3-2	Coordinate System for Converter SNR Analysis	135
4.3-3	Improved PCM to ADM Converter	145
4.3-4	SNR of Improved PCM to ADM Converters for the Limiting Case	158
4.3-5	Correlation Function of w(t) when Input Power Spectrum is White and Bandlimited	159
4.3-6	Correlation Function of w(t) when Input Power Spectrum is Triangular	160
4.3-7	Simulation System for PCM to ADM Converter Performance	162
4.3-8	Test Tones for PCM to ADM Converters and Straight Line Approximations between PCM Points	164
4.3-9	SNR of Staircase Waveform as a Function of the Number of Steps between PCM Points	165
4.3-10	SNR of PCM to ADM Converter with Low Frequency Tone	167
4.3-11	SNR of PCM to ADM Converter with High Frequency Tone	169
4.4-1	Straight Line Approximation between PCM Points and Non-Parametric PCM Midpoint Obtained from Four Adjacent PCM Samples	175
4.4-2	Non-Parametric PCM to ADM Converter with Midpoint Estimate using Four Adjacent PCM Samples	176
4.4-3	SNR of Non-Parametric PCM to ADM Converter with High Frequency Tone	178
4.5-1	Digital Hold Circuit	182
A4-1	Optimum SNR of the Basic PCM to ADM Converter as a Function of Optimizing Time Delay	206

CHAPTER 1

INTRODUCTION

Many modern communication systems utilize digital encoding techniques because of high quality performance and ease in implementation. The rapid advances recently made in the integrated circuit technology contribute greatly to the current interest in digital signal processing. Among the existing encoding techniques, adaptive delta modulation (ADM) and pulse code modulation (PCM) are both very popular and widely used in commercial communications. The simplicity of the DM system makes it very attractive and PCM was the first digital encoding technique, dating back to the late 1940s. Consequently, we shall restrict ourselves to the processing of ADM encoded signals and conversion between ADM and PCM.

Delta modulation is a technique by which an analog signal is encoded into a sequence of binary digits (bits) by periodically comparing the analog signal to an estimate signal. If the error between the analog signal and the estimate is positive, then the bit is +1; if it is negative, then the bit is -1. The estimate, formed from the entire sequence of DM bits, is made to approximate, very closely, the analog signal by increasing or decreasing according to the current bit.

Since the beginning of delta modulation in the early 1950s [1], this simple method of analog-to-digital (A/D)

conversion has undergone many changes. The first DMs constructed and analyzed were composed of analog devices and employed a single "leaky" integrator as their feedback circuit or estimator [2]. To improve dynamic range, the DM feedback circuit soon became continuously adaptive [3-5], that is, the amount of change that the estimator produced each time one bit was transmitted was a function of the past history of the signal. It was not long before the feedback circuit, and then the entire DM, evolved from analog to all-digital devices [6-8]. Currently, we are working with both all-digital linear DMs and various types of all-digital adaptive DMs [9]. Although some analysis and evaluation has been done [10-13], theoretical studies on any system involving ADMs are incomplete, not having treated the nonlinear aspects of the ADMs under investigation.

Throughout this dissertation we confine ourselves to the Song audio mode ADM [14] when evaluating performance of the systems that have been developed. However, the designs are often general enough to be applied to a large class of digital ADMs. All systems in this thesis are designed to be practically realizable. Thus, we only consider operations which can be constructed with standard digital hardware, that is, adders, delays, hard-wired scalars and common logic circuits. Consequently, all our ADM devices can be manufactured with large scale integration (LSI) where the distinct advantage is low cost and high reliability. One objective of the entire thesis is always to work with digitally encoded sig-

nals when developing solutions to our problems.

The three topics considered in this dissertation deal with signals encoded in ADM and PCM formats. The first concentrates on direct arithmetic processing of ADM encoded signals. The next explores conversion from ADM format to PCM format. And the last studies conversion from PCM format to ADM format.

The first topic is investigated because of the popular use of ADM encoded signals and the recent trend toward digital processing of signals. The real objective is to eliminate the need either to demodulate the ADM signal into an analog waveform or change to PCM form before processing can be done. In Ch. 2 we show that the sum, difference and even the product of DM encoded signals can be obtained by operating directly on the serial data, i.e., the ADM bit streams. The sum, difference and product signals are presented in either an ADM or a PCM format. We analyze these processors, discuss their hardware complexity and test them via simulation on a digital computer. The performance is evaluated using a technique developed in Ch. 2. Finally, a comparison is made with equivalent PCM processing systems.

The conversion topics are studied to achieve compatibility between these two widely used encoding techniques. We can also facilitate digital processing of signals encoded in both ADM and PCM formats by devising translation units between the two systems. In Ch. 3, a general technique is presented for converting from ADM to PCM format without first

demodulating the ADM bit stream and returning to the analog domain. The converter utilizes a non-recursive digital filter that is designed with ideal low pass filter characteristics. A frequency domain analysis is developed to explain why this filtering operation is so vital to the performance of this system. The ADM to PCM converter is simulated on a digital computer and a family of performance curves is presented for the case of ADM encoded sinusoidal signals. We also give the performance of an optimal analog conversion system to facilitate comparison.

When considering PCM to ADM conversion we begin by pointing out the conceptual difficulties that arise. In Ch. 4 we explain several ways to circumvent these difficulties and introduce an "ADM signal estimate tree" to illustrate the difficulties and to aid in overcoming them. We design both parametric and non-parametric PCM to ADM converters. The parametric converters utilize the statistics of the input signal in their design and the non-parametric converters are independent of them. A detailed statistical analysis is developed to evaluate the performance of the parametric converters. Extensive simulations are performed on all these systems and the quality of their operation is displayed with families of signal-to-noise ratio (SNR) curves.

The general results of this thesis are very encouraging. The system designs employed to achieve direct ADM processing and conversion between ADM and PCM formats are relatively simple. All the structures that were developed and simulated

could easily be realized with standard digital hardware. The SNR curves for the DM processors are either within 1-2 dB of the SNR curves for equivalent companded PCM systems or are several dB better when the ADM bit rate is lower. Likewise, the performance of the converters comes within 1-2 dB of the performance of optimal conversion systems. Although we have developed satisfactory solutions to the three problems investigated, we have, by no means, exhausted these topics. There is still room for more research, whether it be theoretical analysis or practical experimentation.

CHAPTER 2

DIRECT ARITHMETIC PROCESSING OF ADM ENCODED SIGNALS

Arithmetic processing of digitally encoded signals is traditionally performed on signals that are PCM encoded via standard parallel processing techniques. However, it is becoming increasingly popular to use other digital techniques to encode signals. We would like to avoid the necessity of having to return to a PCM format whenever we must arithmetically process these signals. For the case of DM encoded signals, it is shown that arithmetic processing can be performed by operating directly on the DM bit stream. The direct DM processors can be constructed using standard digital hardware, that is, binary adders, shift registers, exclusive-OR gates and hard-wired scalars. The performance of these devices is shown to be comparable to PCM processors while the hardware complexity is equivalent to, and in some cases less than, that needed for PCM encoded signals.

2.1 The Basic Digital DM

A delta modulator is essentially a very simple device to digitally encode an analog signal. The DM bit stream is obtained by hard limiting the difference between the analog signal and the DM estimate and transmitting a +1 or a -1 every clock period. The DM estimate is a function of all past DM output bits. We shall be concerned with a type of

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

DM which is all-digital in nature. An all-digital DM is one that employs only digital circuitry to produce the DM signal estimate from the DM bit stream.

In Fig. 2.1-1, we show the basic form of a digital DM which is used in a hardware realization. Let us assume that the input signal, $x(t)$, is bandlimited to f_m and the DM is operating well above the Nyquist rate, i.e., $f_s >> 2f_m$, where f_s is the DM clock frequency. The most general mathematical description of a digital DM is given by the following set of equations:

$$e_x(k) = \mathrm{sgn}[\xi_x(k)], \tag{2.1-1}$$

$$\xi_x(k) = x(k) - \hat{x}(k) \tag{2.1-2}$$

and

$$\hat{x}(k) = \hat{x}(k - 1) + S_x(k), \tag{2.1-3}$$

where

$S_x(k) \equiv$ the step size at the kth interval.

Equation (2.1-3) arises because the DM estimate, $\hat{x}(k)$, is the accumulation of the step sizes as shown in Fig. 2.1-1, i.e.,

$$\hat{x}(k) = \sum_{i=-\infty}^{k} S_x(i). \tag{2.1-4}$$

To simplify our discrete signal representation, we have adapted the notation $x(t = kT_s) = x(k)$, where $T_s = 1/f_s$ is the DM clock period.

To completely specify the particular type of digital DM, we must define the step size algorithm used to formulate $S_x(k)$. For the case of a linear or fixed step DM,

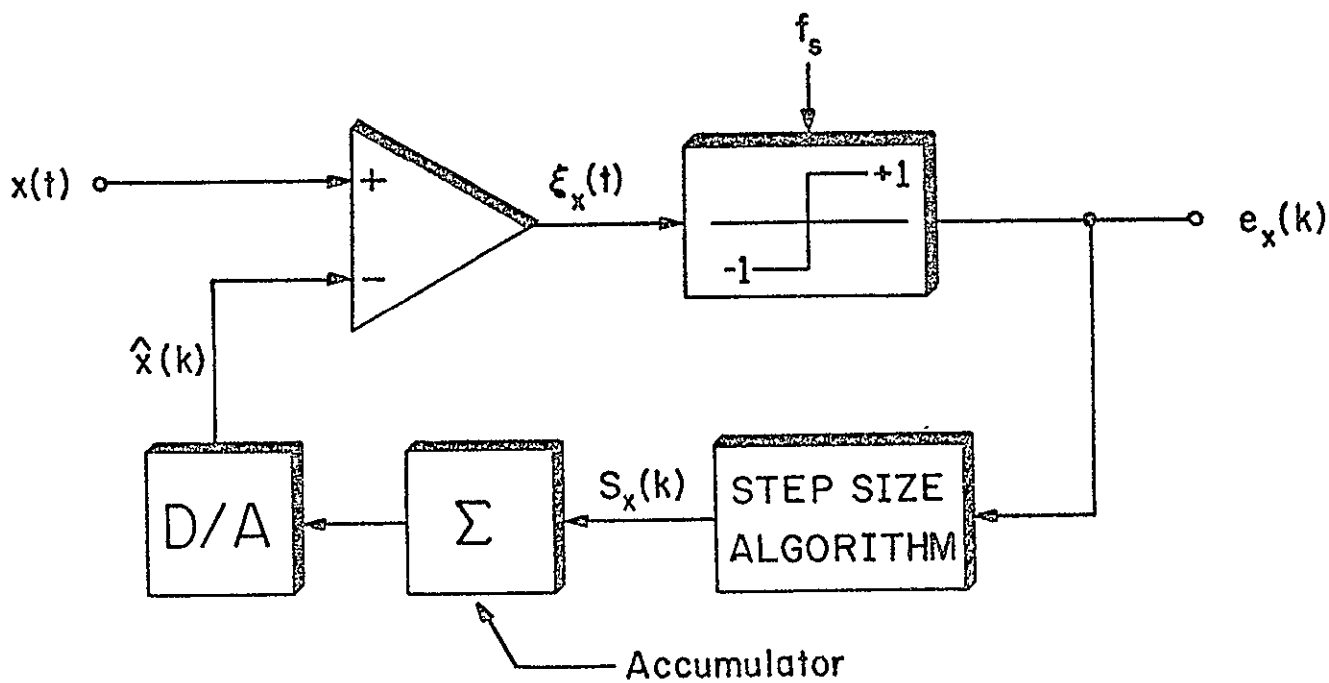


Fig. 2.1-1. Basic Digital Delta Modulator

$$S_x(k) = Se_x(k - 1). \qquad (2.1\text{-}5)$$

For adaptive DMs there are many step size algorithms where the step size adapts to an input signal parameter, generally to its power. We shall be concerned with a class of DMs derived by minimizing a mean square cost function, i.e., those described by the Song Algorithm [15]. In this algorithm, the step size is a function of the two past DM bits and the previous step size. We shall primarily deal with the Song audio mode DM, where the step size changes linearly, i.e.,

$$S_x(k) = |S_x(k - 1)|e_x(k - 1) + Se_x(k - 2), \qquad (2.1\text{-}6)$$

where

$S \equiv$ the magnitude of the minimum step size.

2.2 Direct DM Addition/Subtraction

Consider two signals, $x(t)$ and $y(t)$, both bandlimited to f_m and both DM encoded so that we have only the sequences $\{e_x(k)\}$ and $\{e_y(k)\}$ available. The sum and the difference of these two signals are also bandlimited to f_m. Generally, digital addition (or subtraction) is viewed as the binary sum (or appropriate complementing of the subtrahend and then binary addition) of two K-bit PCM words. If we prohibit overflow, the result is K-bit PCM words at a rate $2f_m$ representing the sum or the difference. Since we are restricted to the DM sequences of $x(t)$ and $y(t)$, we wish to obtain the direct sum of these two signals by performing basic arithmetic processing on the two sequences. To achieve this, we

form the direct sum, $a_D(k)$, as the sum of the individual signal estimates, that is,

$$a_D(k) = \hat{x}(k) + \hat{y}(k). \tag{2.2-1}$$

Using Eq. (2.1-3) for the estimates of $x(t)$ and $y(t)$, we obtain a design equation for the direct sum as a recursive relationship,

$$a_D(k) = a_D(k - 1) + S_x(k) + S_y(k), \tag{2.2-2}$$

where the step sizes are formed directly from the DM bits.

The direct sum, $a_D(k)$, is available in PCM format because the step sizes, used in the design equation, are generated as parallel binary words. To obtain the DM bit stream of the sum, $\{e_a(k)\}$, we merely pass $a_D(k)$ through a digital DM. A block diagram showing the structure for the sum of DM encoded signals is presented in Fig. 2.2-1. The DM digital feedback circuit shown in this figure is constructed with the appropriate step size network followed by an accumulator. Thus, to physically realize the entire DM direct sum system, it requires only a full adder, an accumulator and the necessary step size network.

One way to subtract DM encoded signals is to add the negative of the subtrahend signal. If we wish to form $x(t) - y(t)$, we must change $+S_y(k)$ to $-S_y(k)$ in our design equation. Taking the direct difference, $d_D(k)$, as the difference between the signal estimates, we obtain

$$d_D(k) = d_D(k - 1) + S_x(k) - S_y(k). \tag{2.2-3}$$

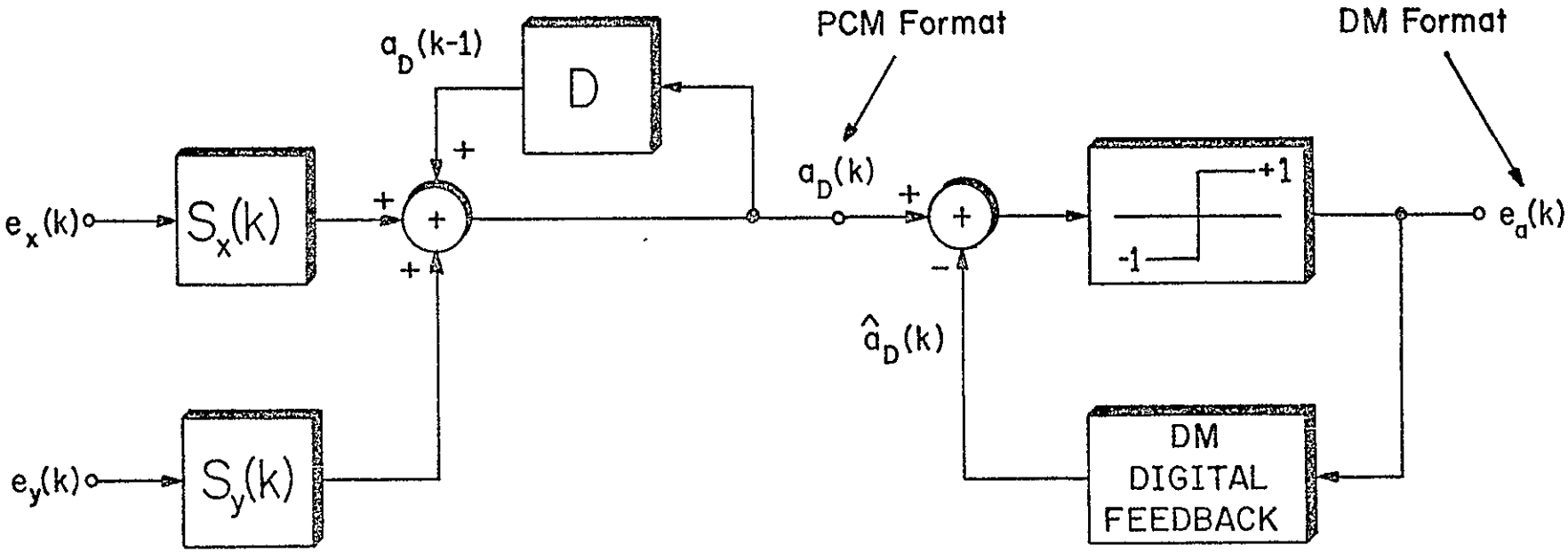


Fig. 2.2-1. Direct Sum of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Thus, the subtraction algorithm has the same structure as the addition algorithm shown in Fig. 2.2-1. Another, even easier method to obtain the difference merely entails inverting each DM bit of the subtrahend signal. This produces $-e_y(k)$ and, for the DM step size algorithm cited above, $-S_y(k)$. Therefore, Fig. 2.2-1 becomes a subtractor by placing an inverter after $e_y(k)$.

The structure derived above is completely independent of the type of DM and is therefore universal for any digital DM definable by Eqs. (2.1-1) through (2.1-3). To realize the direct sum of signals encoded by a particular type of DM, we must construct the circuitry for the step size algorithm employed in the original DM encoder. For the modes cited above, the step size circuitry is constructed with standard digital hardware, i.e., full adders, delays, scalers and exclusive-OR gates. The DM adder for the linear mode is given in Fig. 2.2-2. By applying the Song audio mode algorithm, we can realize the DM adder which is shown in Fig. 2.2-3.

The latter device requires multiplications by $e(k) = \pm 1$ and absolute value operations. Since obtaining the magnitude of a quantity is equivalent to multiplication by +1 if it is positive and by -1 if it is negative, we only need to explain the realization of a multiplication by $e(k)$ using exclusive-OR gates. We shall assume that the internal arithmetic employed is offset binary. For other types of internal arithmetic, similar realizations can be achieved with different logic gates. The main characteristic of offset binary

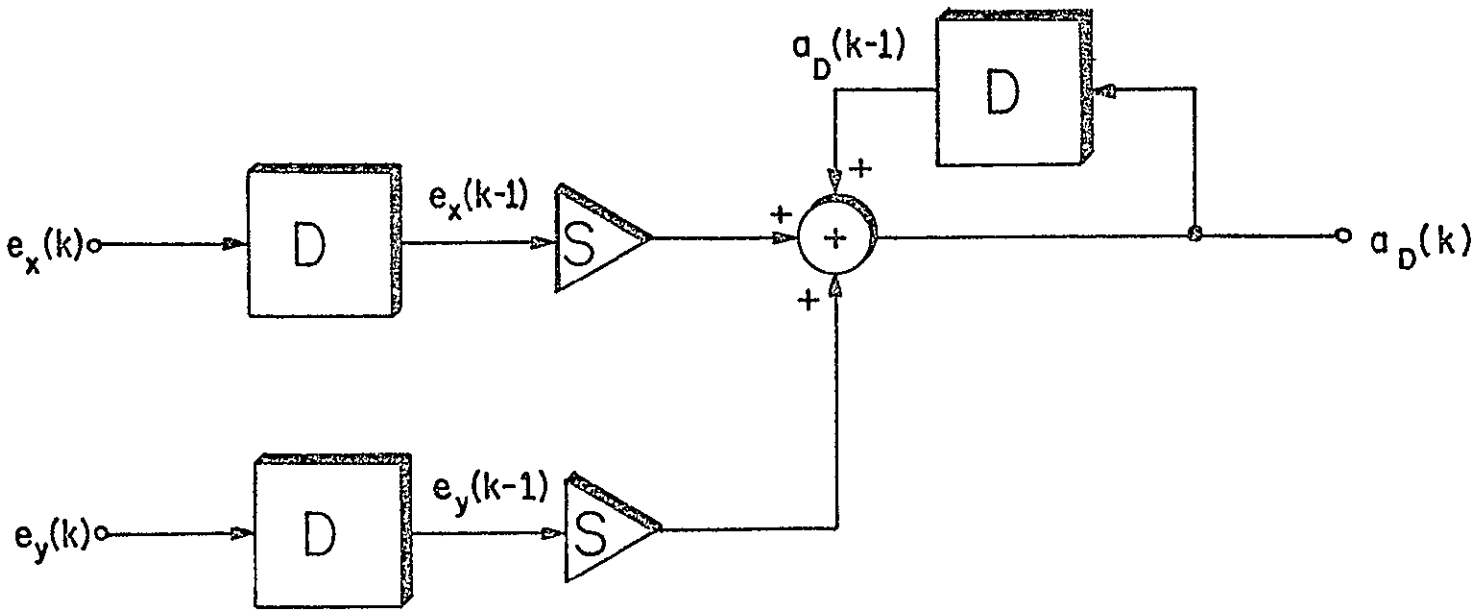


Fig. 2.2-2. DM Adder for Linear Mode

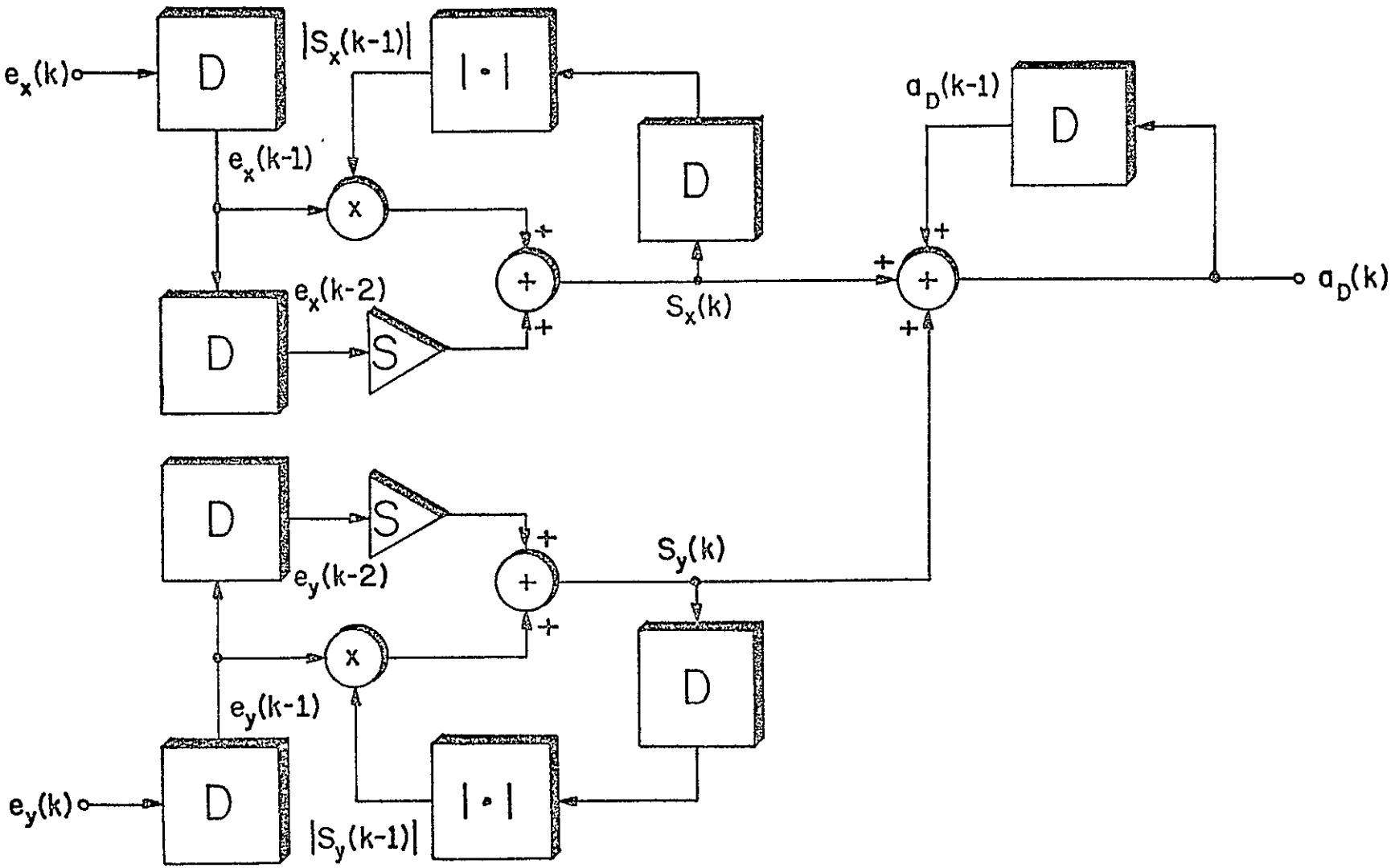


Fig. 2.2-3. DM Adder for Song Audio Mode

arithmetic is its complementary symmetry about the zero axis; that is, the complement of any quantity represents the negative of that quantity. If we multiply by $e(k) = +1$, i.e., a binary 1, we leave the quantity unchanged; and if we multiply by $e(k) = -1$, i.e., a binary 0, we complement the quantity. This operation is easily realized with a bank of exclusive-NOR gates.

In the DM adders, and in all ensuing digital circuits, we shall restrict ourselves to numerical scaling by $I/2^N$, where I and N are positive integers with $I < 2^N$ and N representing the number of bits in our internal arithmetic. Any number of this form is expressable as

$$I/2^N = \sum_{i=1}^{N} a_i/2^i, \qquad (2.2\text{-}4)$$

where

$$a_i = 0 \text{ or } 1.$$

Since a scaling by $1/2^i$ represents a simple shift by i-bits, a scale factor of $I/2^N$ can be hard-wired as a sum of i-bit shifts using only a series of full adders. We emphasize this scaling technique even though multipliers, which will produce the same net result, are readily available on an integrated circuit (IC) chip. The hard-wiring decreases processing time and makes implementation easier because less components are required.

2.3 Averaging Filter

The steady state response of all types of digital DMs to

a constant input is an estimate signal which has been found to exhibit a periodic pattern. For a linear DM, the estimate is a simple square wave with a period of two sampling intervals. For any Song mode DM, the signal estimate is found to exhibit a periodic pattern which repeats every four sampling instants and which is symmetric about the quantized value of the input. In Fig. 2.3-1, we show a typical steady state estimate signal for the Song audio mode. In this figure, the constant input, x, has been quantized to x_q where

$$x_q - S/2 < x < x_q + S/2 \qquad (2.3\text{-}1)$$

and m is a non-negative integer limited by M. If we specify the amplitude range of the DM encoder as V_{pp} and allow the steady state pattern to span this range, then

$$(2M + 1)S \leq V_{pp}. \qquad (2.3\text{-}2a)$$

Consequently,

$$M \leq (V_{pp} - S)/2S. \qquad (2.3\text{-}2b)$$

If the internal arithmetic of the DM encoder has B bits, then the minimum step size is

$$S = V_{pp}/2^B \qquad (2.3\text{-}3)$$

and the upper bound on m can be given as

$$M < 2^{B-1} \quad . \qquad (2.3\text{-}4)$$

Since the direct sum, $a_D(k)$, was formulated as the sum

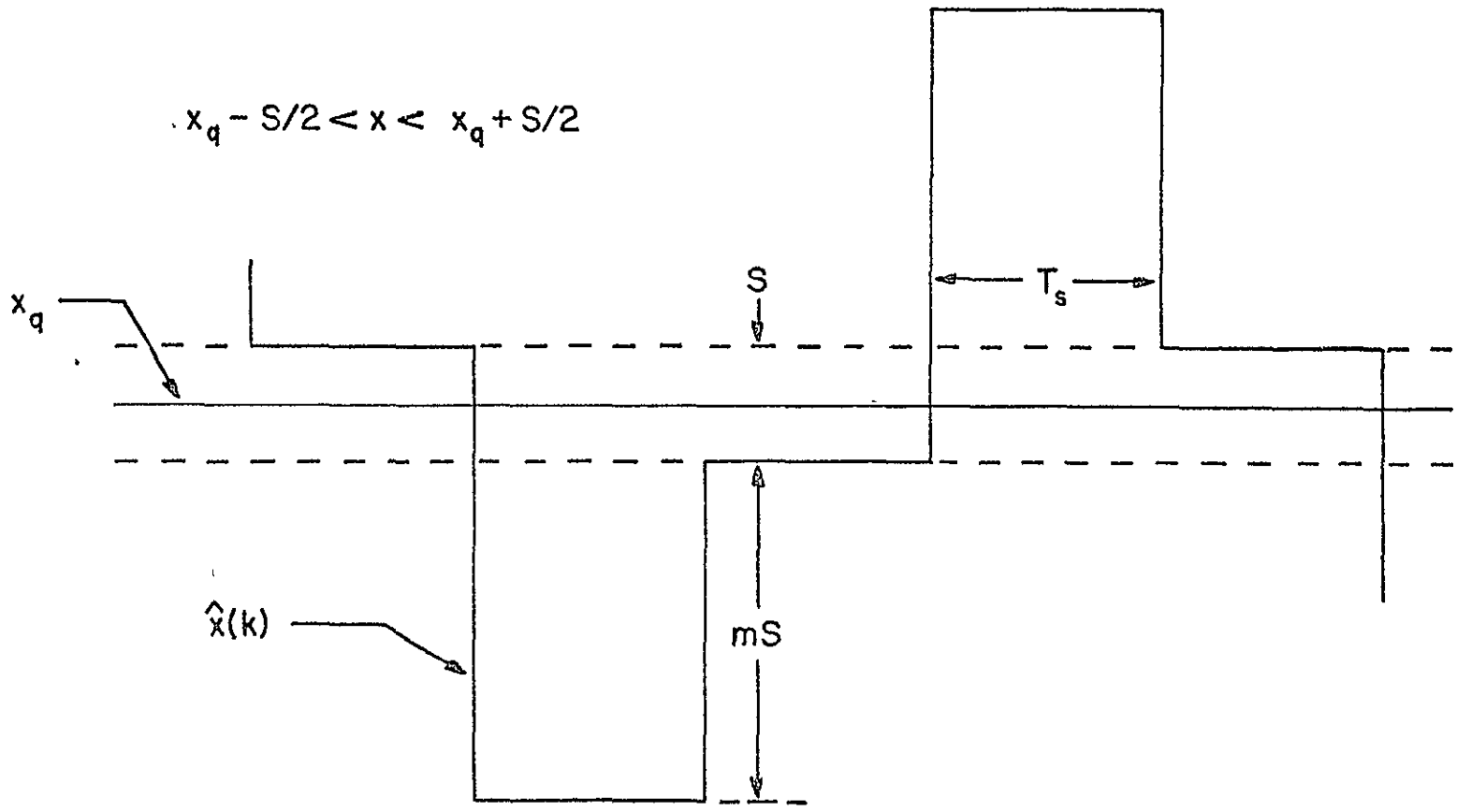


Fig. 2.3-1: Steady State Estimate Signal of the Song Audio Mode DM to a Constant Input

of the individual signal estimates, we expect it to exhibit a periodic pattern when responding to constant inputs. For the Song audio mode, there are four possible steady state direct sum waveforms. A typical waveform, which can represent all four, is given in Fig. 2.3-2. In this figure, both n and r are non-negative integers less than or equal to 2M and y_q is the quantized value of the constant input y where

$$y_q - S/2 < y < y_q + S/2. \qquad (2.3\text{-}5)$$

The important property of the four possible steady state direct sum patterns is that the arithmetic average of any four consecutive values of $a_D(k)$ is always equal to $x_q + y_q$. This fact inspired the use of a four-term averaging filter after $a_D(k)$.

The four-term non-recursive filter that was employed is described by the following equation:

$$A(k) = \tfrac{1}{4}[a_D(k) + a_D(k-1) + a_D(k-2) + a_D(k-3)]. \qquad (2.3\text{-}6)$$

If we apply Eq. (2.3-6) to the waveform given in Fig. 2.3-2, the result is

$$A(k) = x_q + y_q \qquad (2.3\text{-}7)$$

for all k, as long as $a_D(k)$ has reached steady state. Thus, after a four-term averaging filter, the DM sum produces the same result, for constant inputs, obtainable by PCM addition.

To realize this filter, we need not use the structure

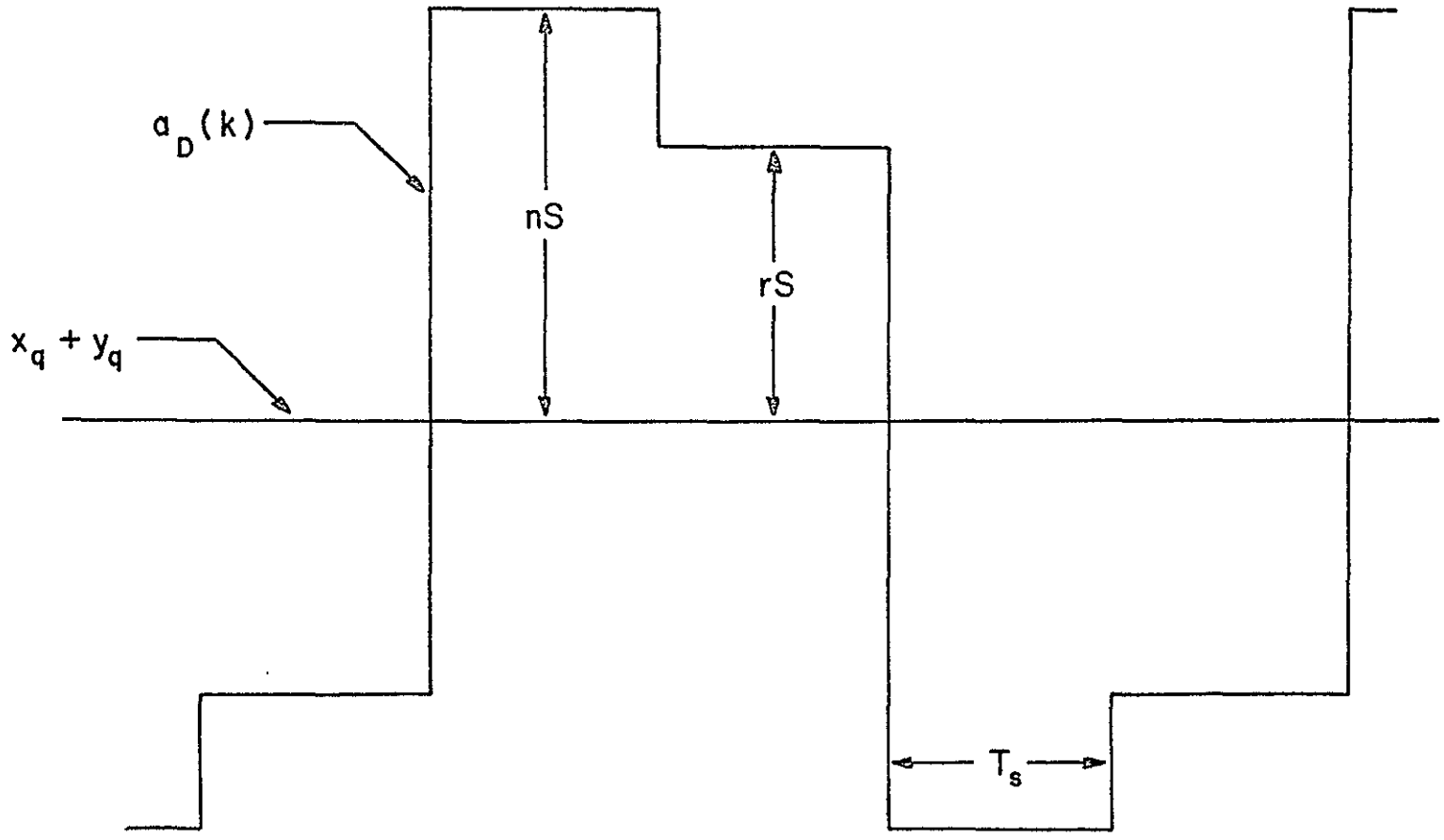


Fig. 2.3-2. Steady State Direct Sum for the Song Audio Mode DM with Constant Inputs

dictated by Eq. (2.3-6). We can save hardware by employing the realization shown in Fig. (2.3-3) which uses two, rather than three, adders. In general, by extending this structure, it is possible to realize any N-term averaging filter, where $N = 2^t$ and t is a positive integer, in a similar fashion. The advantage is that, instead of needing $2^t - 1$ adders in the realization, only t adders are required.

To illustrate the function of the four-term averaging filter, we shall determine its digital transfer function, $H_a(z)$. Assuming zero initial conditions and taking the Z-transform of Eq. (2.3-6), we obtain

$$H_a(z) = \frac{A(z)}{a_D(z)} = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3}). \qquad (2.3\text{-}8)$$

To display the frequency characteristics of this filter, we let $z = \exp(j\omega T_s)$ and find that

$$|H_a(\omega)| = |\cos(\omega T_s/2)\cos(\omega T_s)|. \qquad (2.3\text{-}9)$$

In Fig. (2.3-4), we plot $|H_a(\omega)|$ on an abscissa normalized to f_s. A complete derivation of Eq. (2.3-9) is given in App. 1.

From Fig. (2.3-4) and Eq. (2.3-9), we observe that $H_a(\omega)$ has zeros at integer multiples of $f_s/4$ when the integer is divisible by 4. It is precisely the first zero that eliminates the four-sampling-interval periodic component in the direct sum. The four-term averaging filter exhibits low pass filter (LPF) characteristics and even slightly attenu-

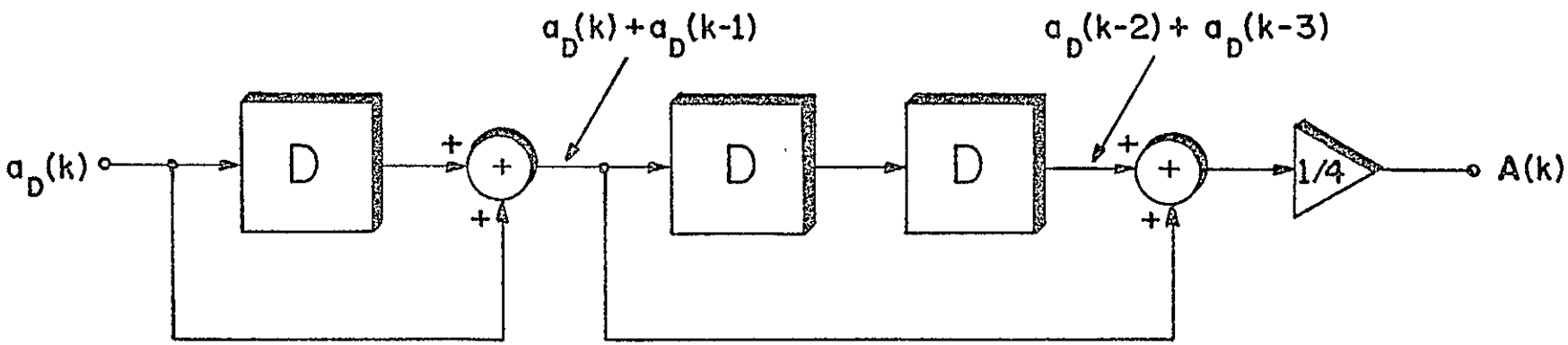


Fig. 2.3-3. Four-Term Averaging Filter Realization

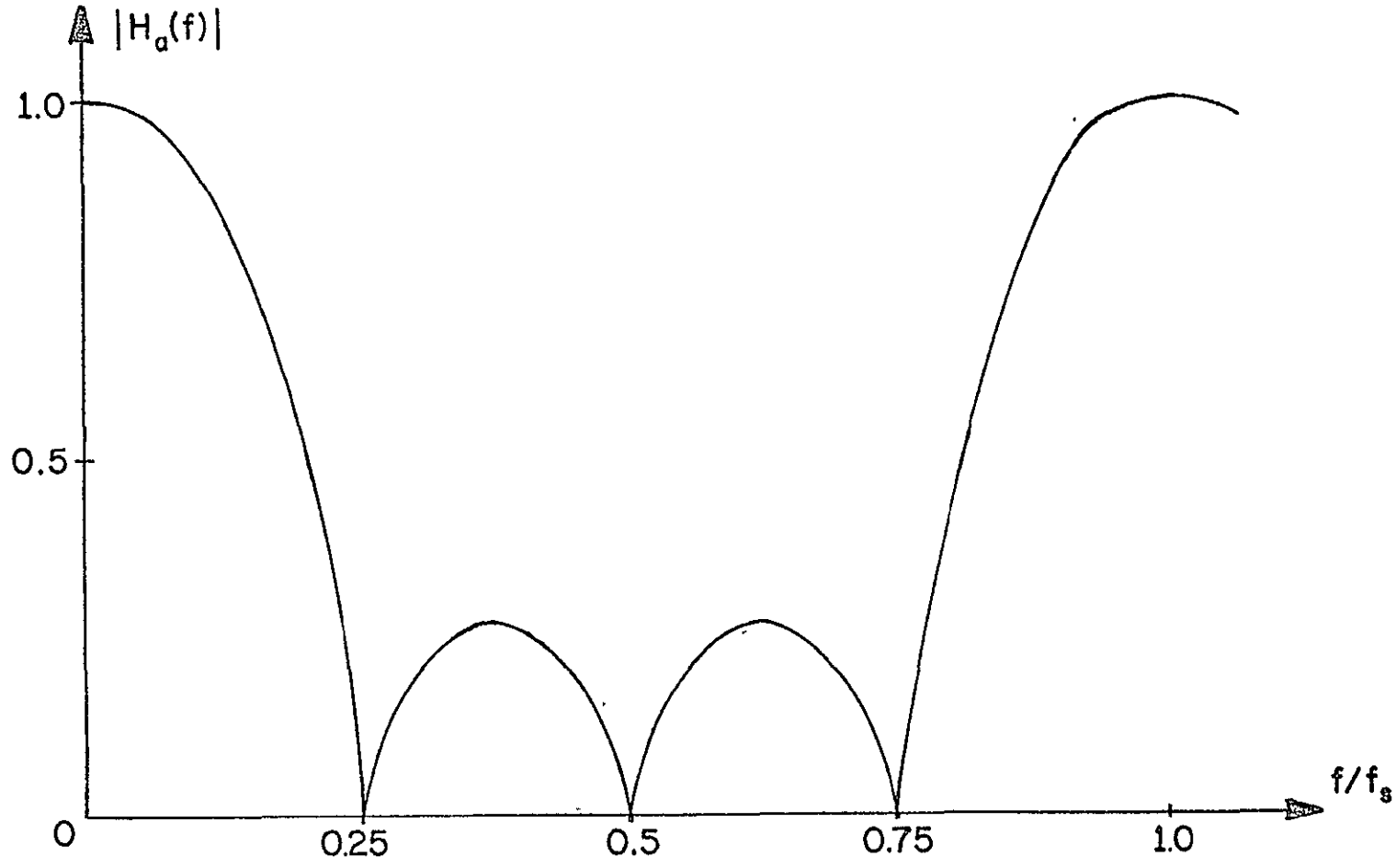


Fig. 2.3-4. Amplitude-Frequency Characteristics of the Four-Term Non-Recursive Averaging Filter

ates baseband frequencies. Thus, care must be taken in utilizing this filter because it can introduce some distortion to baseband signals. However, if the maximum baseband frequency is in the area of $f_s/10$, then distortion will be minimal since $|H_a(f = f_s/10)| = 0.77$, which is well above the 3 dB point of this filter. This is not an unreasonable requirement because the audio mode DM generally operates at f_s = 32K bits/second and speech is usually bandlimited to 2500 to 3500 Hz.

2.4 Direct DM Multiplication

Traditionally digital multiplication is treated as a static operation; that is, two K-bit PCM words are either fed into a combinatoric circuit or into a read only memory (ROM) that has been built to perform the multiplication operation. We could even use a random access memory (RAM) that has been preset to multiply, with the appropriate input logic circuit. The result is the product in the form of a 2K-bit PCM word. There are, of course, dynamic techniques capable of performing digital multiplication and we shall discuss them in connection with hardware complexity. In either case, if we wish to multiply two signals, both bandlimited to f_m, then we must perform the static or dynamic operation on PCM words from each signal at a rate of $4f_m$ to obtain the product in a PCM format. This is because the product will be bandlimited to $2f_m$ as we know from the convolution theorem. This point is important for our later

performance comparison with PCM. There are other ways to obtain the product in PCM form by using the samples of both signals at a rate of $2f_m$. This matter is further pursued in App. 2.

The problem that we consider here is the formation of the product of $x(t)$ and $y(t)$ when we only have their DM sequences, $\{e_x(k)\}$ and $\{e_y(k)\}$, available. Once again we restrict ourselves to a design structure that can be implemented with standard digital hardware. Forming the direct product as the product of the individual signal estimates, we have

$$p_D(k) = \hat{x}(k)\hat{y}(k). \tag{2.4-1}$$

As in the case of the direct sum, we can develop a recursive relationship as follows:

$$\begin{aligned} p_D(k) = p_D(k-1) &+ S_y(k)\hat{x}(k-1) + S_x(k)\hat{y}(k-1) \\ &+ S_x(k)S_y(k). \end{aligned} \tag{2.4-2}$$

The basic block diagram showing the direct product, in PCM format $[p_D(k)]$ and in DM format $[e_p(k)]$, is given in Fig. (2.4-1). Although the structure for the direct product is universal for any digital DM definable by Eqs. (2.1-1) through (2.1-3), it will be useful only if the step size algorithm is such that we can recursively realize the particle products, that is, the last three terms in Eq. (2.4-2).

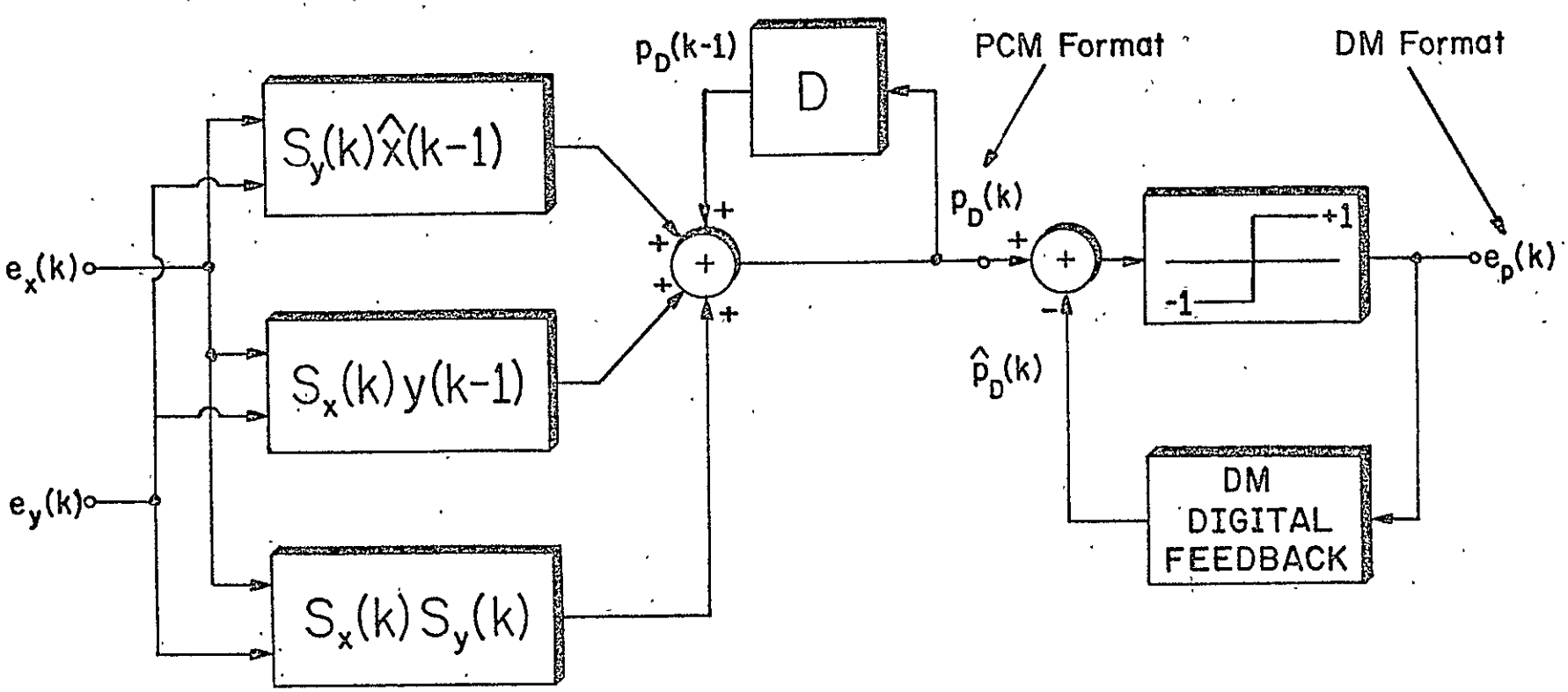


Fig. 2.4-1. Direct Product of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

For the linear DM, no difficulty arises and the direct product is

$$p_D(k) = p_D(k-1) + Se_y(k-1)\hat{x}(k-1) + Se_x(k-1)\hat{y}(k-1) + S^2 e_x(k-1)e_y(k-1). \tag{2.4-3}$$

The realization of this system, shown in Fig. (2.4-2), is extremely easy because there are no non-linear operations, only simple scaling including multiplication by +1 or -1.

To derive the recursive relationships for the partial products with the Song audio mode algorithm, we must use a step size relationship common to all types of DMs, that is,

$$S_x(k) = |S_x(k)|e_x(k-1). \tag{2.4-4}$$

This equation says that the sign of the present step size, $S_x(k)$, is dictated by the past DM output bit, $e_x(k-1)$. From Eq. (2.1-6), we see that this property is applicable for the Song audio mode as long as $|S_x(k-1)| \geq S$. If $S_x(k-1) = 0$ and $e_x(k-1) = e_x(k-2)$, then this property is also valid. Only when $S_x(k-1) = 0$ and $e_x(k-1) \neq e_x(k-2)$, the step size relationship becomes invalid. This invalidity is caused by a hardware limitation that allows the step size to be zero rather than an arbitrarily small value. We shall show, however, that the condition of invalidity has a very low probability of occurrence and the resulting signal estimate used in the multiplication algorithm does not substantially degrade the product.

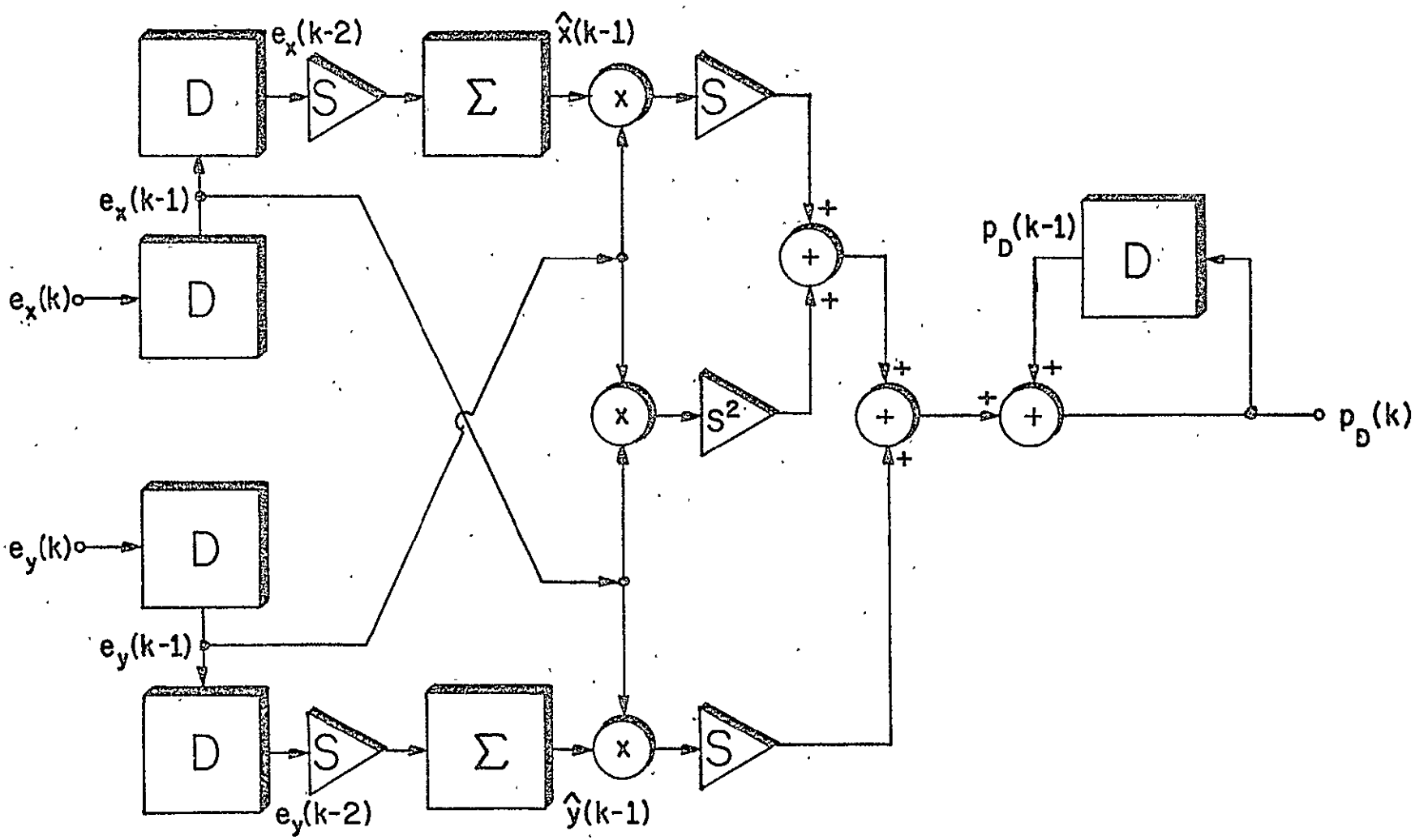


Fig. 2.4-2. DM Multiplier for Linear Mode

Equation (2.4-4) can only be invalid when $S_x(k) = 0$. A zero step size occurs primarily when the ADM is in its minimum steady state pattern. That is, the estimate resembles Fig. 2.3-1 when m = 0. This corresponds to the audio signal being zero because, in speech, 50% of the time there is no voice. Recently, step size statistics have been obtained for the Song audio mode ADM, using actual speech signals. They show that the probability of a zero step size is approximately 0.04 when f_s = 32K bits/second. Since $S_x(k) = 0$ occurs twice in a minimum steady state estimate pattern, the probability of such a pattern is 0.08. Let us assume that the audio signal is equally likely to increase or decrease from its zero value in any of the four periods of the steady state pattern. Only 2 of the 8 signal variations give rise to the condition when Eq. (2.4-4) will be invalid. Therefore, the probability of invalidity is 0.02.

We have created a situation in Fig. 2.4-3 where the step size relationship is invalid. The solid curve represents the true ADM estimate and the broken line waveform is the estimate used in the multiplication algorithm. From this figure, we observe that the estimate used in the multiplication algorithm is just as good an approximation to the audio signal as the true estimate. We shall see, in Sec. 2.8.3, that the use of Eq. (2.4-2) does not noticeably effect the output SNR of the DM multiplier.

Now that we have justified the step size relationship, we can use it to express recursively the partial products

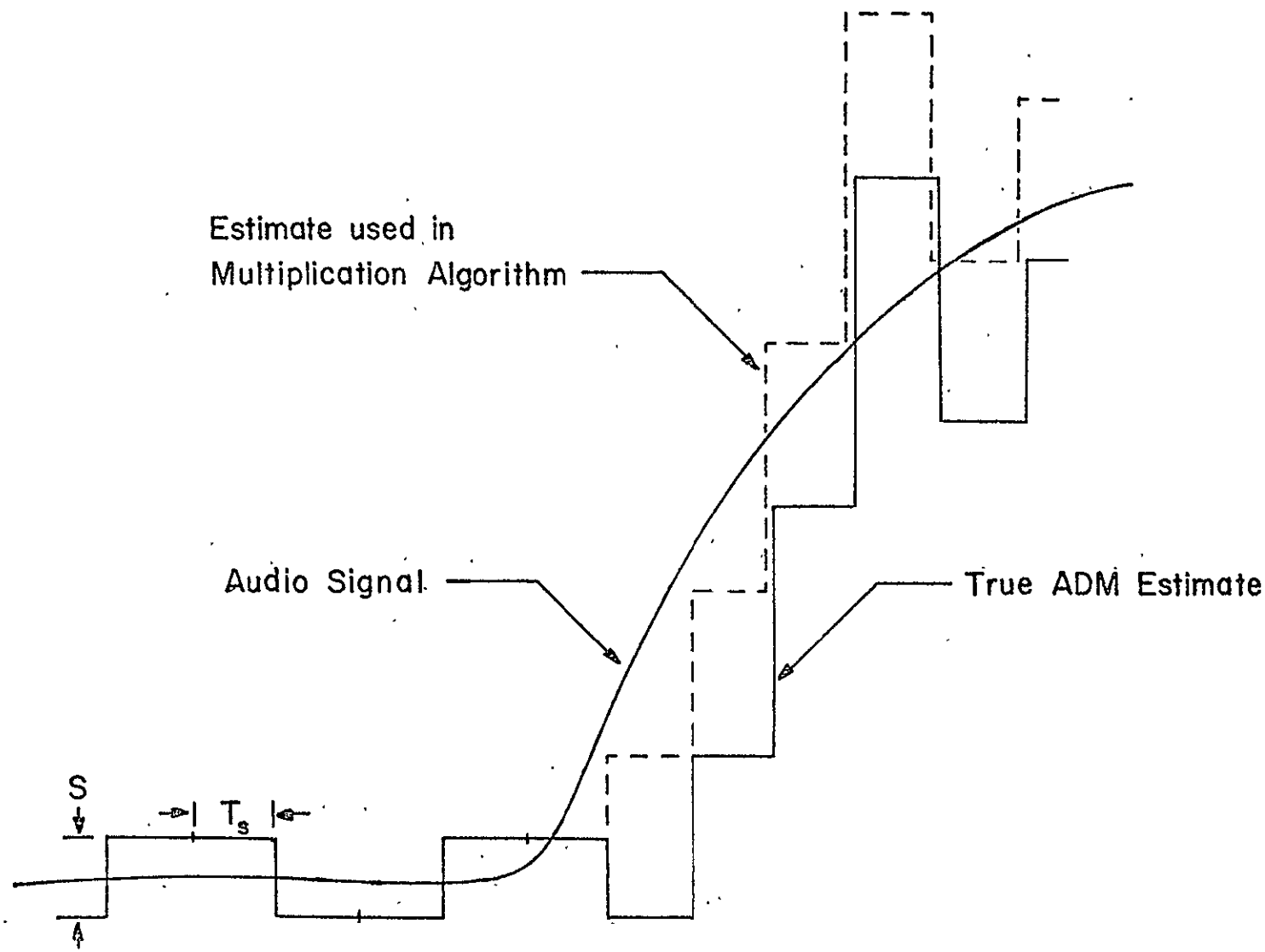


Fig. 2.4-3. Example when the Step Size Relationship is Invalid

for the Song audio mode:

$$S_y(k)\hat{x}(k-1) = S_y(k-1)x(k-2)e_y(k-1)e_y(k-2)$$
$$+ S_x(k-1)S_y(k-1)e_y(k-1)e_y(k-2) \qquad (2.4\text{-}5)$$
$$+ \hat{x}(k-1)Se_y(k-2),$$

$$S_x(k)\hat{y}(k-1) = S_x(k-1)\hat{y}(k-2)e_x(k-1)e_x(k-2)$$
$$+ S_y(k-1)S_x(k-1)e_x(k-1)e_x(k-2) \qquad (2.4\text{-}6)$$
$$+ \hat{y}(k-1)Se_x(k-2),$$

$$S_x(k)S_y(k) = |S_x(k-1)S_y(k-1)|e_x(k-1)e_y(k-1)$$
$$+ S|S_x(k-1)|e_x(k-1)e_y(k-2)$$
$$+ S|S_y(k-1)|e_y(k-1)e_x(k-2) \qquad (2.4\text{-}7)$$
$$+ S^2e_x(k-2)e_y(k-2).$$

Equations (2.4-5), (2.4-6) and (2.4-7) are readily realizable with standard digital hardware similar to the DM adder shown in Fig. 2.2-3. These three terms can be constructed with nothing more complicated than adders, delays, hard-wired scalers and exclusive-OR gates to multiply by ±1 and produce the absolute value.

All of the design structures that we have derived are accumulator type systems. For both the adder (Sec. 2.3) and the multiplier (Sec. 2.4), for all DM modes, the present out-

put is equal to the past output plus additional terms. Thus, it is important to begin with the correct initial condition for the past output, or else suffer a constant offset error. It is convenient to start with both signals, x(t) and y(t), at zero so that we can employ a zero initial condition for the past output.

As in the case of the direct sum, we expect the direct product, since it is formulated as the product of the individual signal estimates, to exhibit a periodic pattern when responding to constant inputs. When using the Song audio mode algorithm, the direct product generates four possible steady state waveforms. In Fig. 2.4-4, we show the general structure of a steady state waveform. The values of C_1 and C_2 depend upon x_q and y_q and the amplitude of the steady state error pattern ($\hat{x} - x_q$ and $\hat{y} - y_q$), while d_1 and d_2 depend only on the latter of these two. The numerical values of d_1 and d_2 can be entirely different, but they both have the same form, as can be seen by multiplying two steady state patterns together, that is,

$$|d_i| = L/4, \tag{2.4-8}$$

where

$$i = 1, 2,$$

$$L = \text{a positive integer}$$

and

$$L \leq 4M^2 + 4M + 1 \tag{2.4-9}$$

where M is bounded in Eq. (2.3-4).

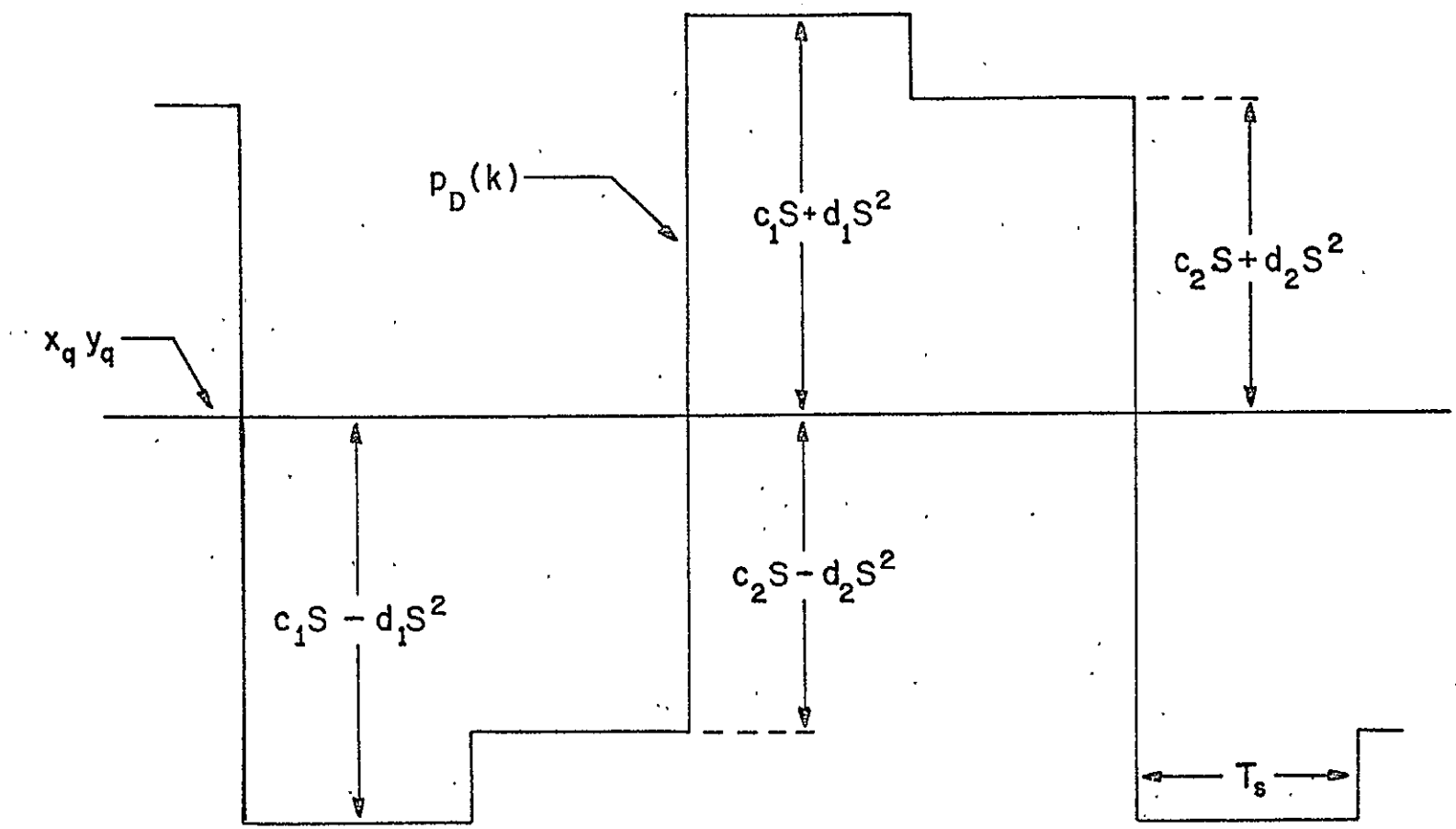


Fig. 2.4-4. Steady State Direct Product for Song Audio Mode DM with Constant Inputs

The arithmetic average of any four consecutive values of $p_D(k)$ always equals the product of the quantized values of the inputs plus a second-order term depending on S^2. This warrants the use of the following four-term non-recursive filter after $p_D(k)$:

$$P(k) = \frac{1}{4}[p_D(k) + p_D(k-1) + p_D(k-2) + p_D(k-3)]. \quad (2.4\text{-}10)$$

Applying Eq. (2.4-10) to the waveform shown in Fig. 2.4-3, we see that

$$P(k) = x_q y_q + (d_1 + d_2)S^2/2 \quad (2.4\text{-}11)$$

for all k, as long as $p_D(k)$ has reached the steady state. We have found, through computer simulations, that the factor $(d_1 + d_2)/2$ generally is no larger than 10 or 20. In a practical DM encoder with 10 bits of internal arithmetic and an amplitude range of $V_{pp} = 10$ volts, the minimum step size, S, will be approximately 10 millivolts. Therefore, the second-order term will be in the order of 1-2 millivolts. Even if we allow $(d_1 + d_2)/2$ to be 100, the error only reaches 10 millivolts or one minimum step size. Certainly, one step size out of 1024 can be considered insignificant. For any reasonably small value of step size, the second order term, $(d_1 + d_2)S^2/2$, is negligible and thus, after the four-term averaging filter, the DM product yields a result almost identical to the PCM product.

The direct DM product design structure that we have derived is not a unique solution to this problem. The de-

performance comparison with PCM. There are other ways to obtain the product in PCM form by using the samples of both signals at a rate of $2f_m$. This matter is further pursued in App. 2.

The problem that we consider here is the formation of the product of $x(t)$ and $y(t)$ when we only have their DM sequences, $\{e_x(k)\}$ and $\{e_y(k)\}$, available. Once again we restrict ourselves to a design structure that can be implemented with standard digital hardware. Forming the direct product as the product of the individual signal estimates, we have

$$p_D(k) = \hat{x}(k)\hat{y}(k). \tag{2.4-1}$$

As in the case of the direct sum, we can develop a recursive relationship as follows:

$$\begin{aligned} p_D(k) = p_D(k-1) &+ S_y(k)\hat{x}(k-1) + S_x(k)\hat{y}(k-1) \\ &+ S_x(k)S_y(k). \end{aligned} \tag{2.4-2}$$

The basic block diagram showing the direct product, in PCM format $[p_D(k)]$ and in DM format $[e_p(k)]$, is given in Fig. (2.4-1). Although the structure for the direct product is universal for any digital DM definable by Eqs. (2.1-1) through (2.1-3), it will be useful only if the step size algorithm is such that we can recursively realize the particle products, that is, the last three terms in Eq. (2.4-2).

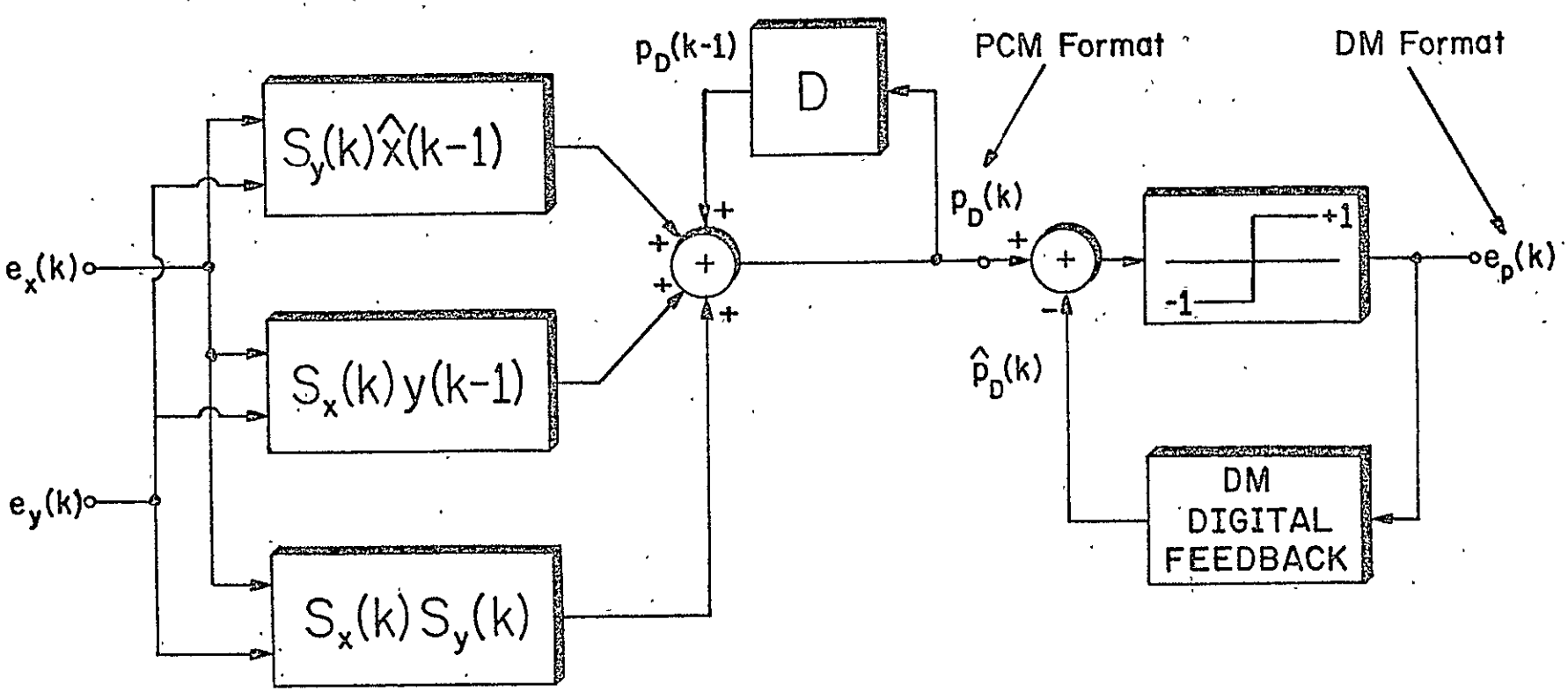


Fig. 2.4-1. Direct Product of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

For the linear DM, no difficulty arises and the direct product is

$$p_D(k) = p_D(k-1) + Se_y(k-1)\hat{x}(k-1) + Se_x(k-1)\hat{y}(k-1) + S^2 e_x(k-1)e_y(k-1). \tag{2.4-3}$$

The realization of this system, shown in Fig. (2.4-2), is extremely easy because there are no non-linear operations, only simple scaling including multiplication by +1 or -1.

To derive the recursive relationships for the partial products with the Song audio mode algorithm, we must use a step size relationship common to all types of DMs, that is,

$$S_x(k) = |S_x(k)|e_x(k-1). \tag{2.4-4}$$

This equation says that the sign of the present step size, $S_x(k)$, is dictated by the past DM output bit, $e_x(k-1)$. From Eq. (2.1-6), we see that this property is applicable for the Song audio mode as long as $|S_x(k-1)| \geq S$. If $S_x(k-1) = 0$ and $e_x(k-1) = e_x(k-2)$, then this property is also valid. Only when $S_x(k-1) = 0$ and $e_x(k-1) \neq e_x(k-2)$, the step size relationship becomes invalid. This invalidity is caused by a hardware limitation that allows the step size to be zero rather than an arbitrarily small value. We shall show, however, that the condition of invalidity has a very low probability of occurrence and the resulting signal estimate used in the multiplication algorithm does not substantially degrade the product.

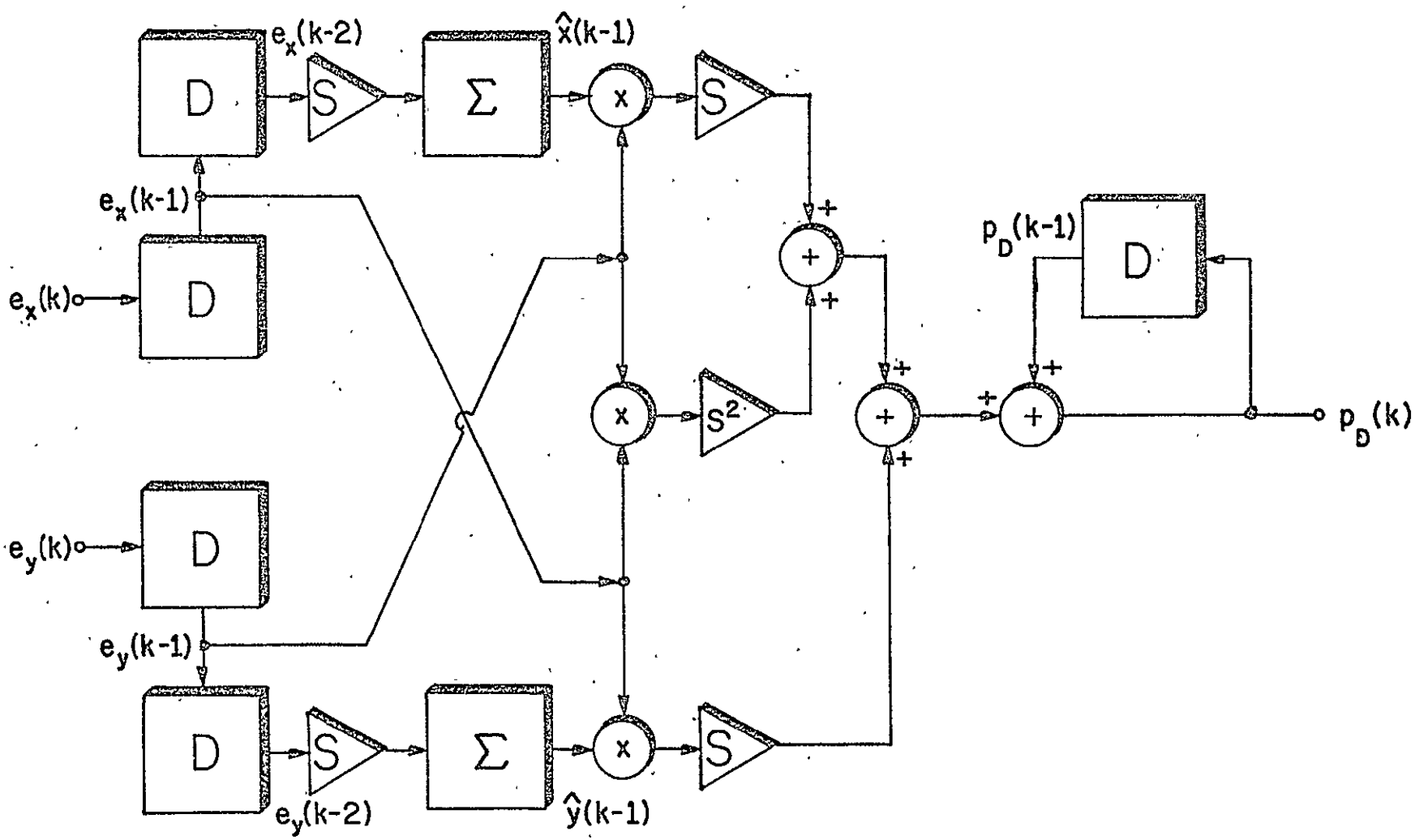


Fig. 2.4-2. DM Multiplier for Linear Mode

Equation (2.4-4) can only be invalid when $S_x(k) = 0$. A zero step size occurs primarily when the ADM is in its minimum steady state pattern. That is, the estimate resembles Fig. 2.3-1 when m = 0. This corresponds to the audio signal being zero because, in speech, 50% of the time there is no voice. Recently, step size statistics have been obtained for the Song audio mode ADM, using actual speech signals. They show that the probability of a zero step size is approximately 0.04 when f_s = 32K bits/second. Since $S_x(k) = 0$ occurs twice in a minimum steady state estimate pattern, the probability of such a pattern is 0.08. Let us assume that the audio signal is equally likely to increase or decrease from its zero value in any of the four periods of the steady state pattern. Only 2 of the 8 signal variations give rise to the condition when Eq. (2.4-4) will be invalid. Therefore, the probability of invalidity is 0.02.

We have created a situation in Fig. 2.4-3 where the step size relationship is invalid. The solid curve represents the true ADM estimate and the broken line waveform is the estimate used in the multiplication algorithm. From this figure, we observe that the estimate used in the multiplication algorithm is just as good an approximation to the audio signal as the true estimate. We shall see, in Sec. 2.8.3, that the use of Eq. (2.4-2) does not noticeably effect the output SNR of the DM multiplier.

Now that we have justified the step size relationship, we can use it to express recursively the partial products

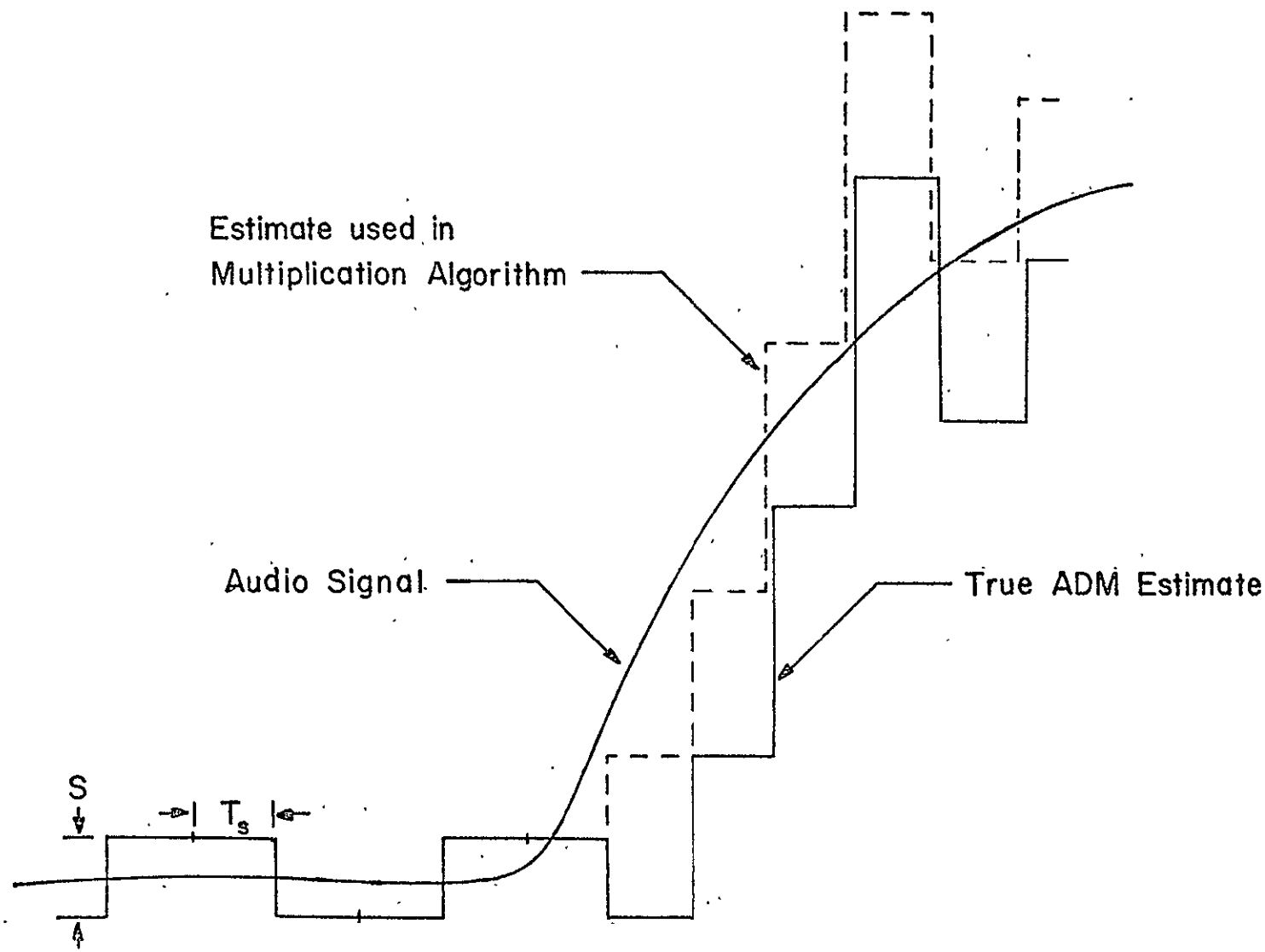


Fig. 2.4-3. Example when the Step Size Relationship is Invalid

for the Song audio mode:

$$S_y(k)\hat{x}(k-1) = S_y(k-1)x(k-2)e_y(k-1)e_y(k-2)$$
$$+ S_x(k-1)S_y(k-1)e_y(k-1)e_y(k-2) \qquad (2.4\text{-}5)$$
$$+ \hat{x}(k-1)Se_y(k-2),$$

$$S_x(k)\hat{y}(k-1) = S_x(k-1)\hat{y}(k-2)e_x(k-1)e_x(k-2)$$
$$+ S_y(k-1)S_x(k-1)e_x(k-1)e_x(k-2) \qquad (2.4\text{-}6)$$
$$+ \hat{y}(k-1)Se_x(k-2),$$

$$S_x(k)S_y(k) = |S_x(k-1)S_y(k-1)|e_x(k-1)e_y(k-1)$$
$$+ S|S_x(k-1)|e_x(k-1)e_y(k-2)$$
$$+ S|S_y(k-1)|e_y(k-1)e_x(k-2) \qquad (2.4\text{-}7)$$
$$+ S^2e_x(k-2)e_y(k-2).$$

Equations (2.4-5), (2.4-6) and (2.4-7) are readily realizable with standard digital hardware similar to the DM adder shown in Fig. 2.2-3. These three terms can be constructed with nothing more complicated than adders, delays, hard-wired scalers and exclusive-OR gates to multiply by ±1 and produce the absolute value.

All of the design structures that we have derived are accumulator type systems. For both the adder (Sec. 2.3) and the multiplier (Sec. 2.4), for all DM modes, the present out-

put is equal to the past output plus additional terms. Thus, it is important to begin with the correct initial condition for the past output, or else suffer a constant offset error. It is convenient to start with both signals, x(t) and y(t), at zero so that we can employ a zero initial condition for the past output.

As in the case of the direct sum, we expect the direct product, since it is formulated as the product of the individual signal estimates, to exhibit a periodic pattern when responding to constant inputs. When using the Song audio mode algorithm, the direct product generates four possible steady state waveforms. In Fig. 2.4-4, we show the general structure of a steady state waveform. The values of C_1 and C_2 depend upon x_q and y_q and the amplitude of the steady state error pattern ($\hat{x} - x_q$ and $\hat{y} - y_q$), while d_1 and d_2 depend only on the latter of these two. The numerical values of d_1 and d_2 can be entirely different, but they both have the same form, as can be seen by multiplying two steady state patterns together, that is,

$$|d_i| = L/4, \tag{2.4-8}$$

where

$$i = 1, 2,$$

$$L = \text{a positive integer}$$

and

$$L \leq 4M^2 + 4M + 1 \tag{2.4-9}$$

where M is bounded in Eq. (2.3-4).

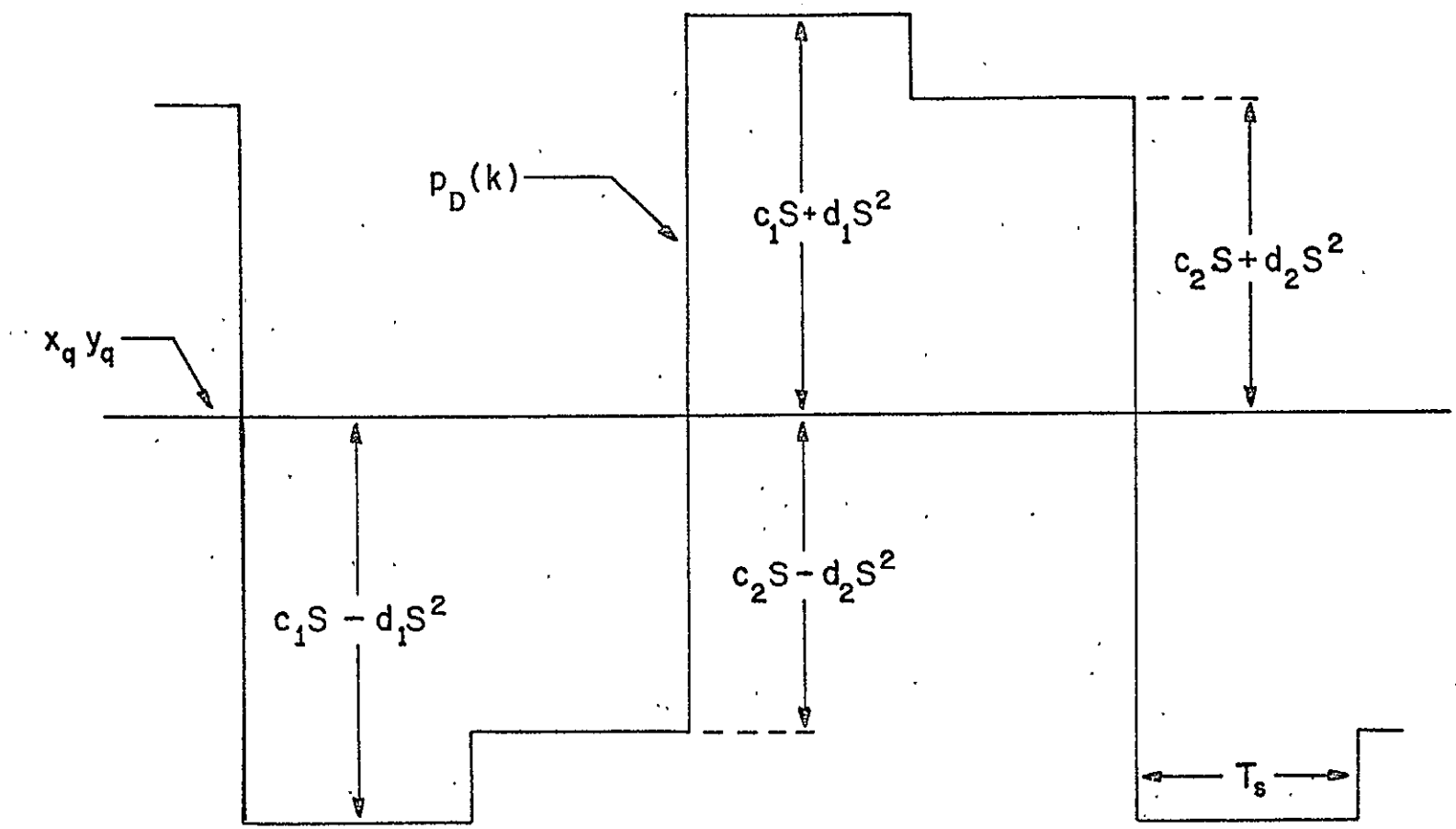


Fig. 2.4-4. Steady State Direct Product for Song Audio Mode DM with Constant Inputs

The arithmetic average of any four consecutive values of $p_D(k)$ always equals the product of the quantized values of the inputs plus a second-order term depending on S^2. This warrants the use of the following four-term non-recursive filter after $p_D(k)$:

$$P(k) = \frac{1}{4}[p_D(k) + p_D(k-1) + p_D(k-2) + p_D(k-3)]. \quad (2.4\text{-}10)$$

Applying Eq. (2.4-10) to the waveform shown in Fig. 2.4-3, we see that

$$P(k) = x_q y_q + (d_1 + d_2)S^2/2 \quad (2.4\text{-}11)$$

for all k, as long as $p_D(k)$ has reached the steady state. We have found, through computer simulations, that the factor $(d_1 + d_2)/2$ generally is no larger than 10 or 20. In a practical DM encoder with 10 bits of internal arithmetic and an amplitude range of $V_{pp} = 10$ volts, the minimum step size, S, will be approximately 10 millivolts. Therefore, the second-order term will be in the order of 1-2 millivolts. Even if we allow $(d_1 + d_2)/2$ to be 100, the error only reaches 10 millivolts or one minimum step size. Certainly, one step size out of 1024 can be considered insignificant. For any reasonably small value of step size, the second order term, $(d_1 + d_2)S^2/2$, is negligible and thus, after the four-term averaging filter, the DM product yields a result almost identical to the PCM product.

The direct DM product design structure that we have derived is not a unique solution to this problem. The de-

sign presented does, however, perform well and this will be seen from the simulation responses for elementary input waveforms and also from the SNR performance curves. There are other design techniques that could have been incorporated into the direct product design. We could introduce a "leak" factor in Eq. (2.4-2) and feedback a fraction of the output, $p_D(k)$, to generate, say $0.9p_D(k - 1)$. Alternately, we might use a different averaging filter after $p_D(k)$; or we could perform some kind of averaging on $\hat{x}(k)$ and $\hat{y}(k)$ before we form their product. Each of these ideas would have to be analyzed individually to determine its merits and shortcomings in formulating the direct DM product.

2.5 Hardware Complexity

From Eq. (2.2-2) or Fig. 2.2-1 we see that the complexity and quantity of the hardware needed for a DM adder is essentially equivalent to that needed for a PCM adder. In PCM addition, since we have two K-bit words coming from $x(t)$ and $y(t)$, we require two K-bit input storage registers, one K-bit full adder and, since we do not allow overflow, one K-bit output register. To obtain the DM direct sum, we need the step size circuitry for both signals (some of which can be time shared) terminating in registers with less than K-bit capacity, one transfer register with enough bits to represent twice the maximum step size, one K-bit full adder and one K-bit delay register.

The comparison of hardware complexity for multiplica-

tion is somewhat more involved. PCM multiplication is generally treated as a static operation where two K-bit words are either fed into a combinatorial circuit, or into a preprogrammed ROM, or into a repeating add-store-and-shift circuit. We must also remember that to multiply two signals, bandlimited to f_m, we must perform this static operation on the PCM words from the two signals at a rate of $4f_m$ since the product will be bandlimited to $2f_m$.

Now we can examine the hardware complexity needed for these PCM multipliers. A combinatorial circuit needs two K-bit input storage registers, K^2 AND gates, K K-bit full adders and a 2K-bit output storage register. This is easily seen by observing the structure that arises when we use "long" multiplication to obtain the product of two K-bit words, $A_K \cdots A_2 A_1$ and $B_K \cdots B_2 B_1$. A ROM, with a 2K-bit input address, normally has 2^{2K} memory locations. Even though there are only K^2 different product values, the ROM must still use 2^{2K} memory locations to multiply as well as needing input and output registers. The repeating add-store-and-shift device requires two K-bit input registers, a K-bit shift register, K AND gates, a 2K-bit full adder and a 2K-bit output register. However, for this last multiplier, we must perform K repeated additions in the time interval $1/4f_m$ before we obtain the final product word.

Considering the DM multiplier for the linear mode, as shown in Fig. 2.4-2, the hardware needed is two 2-bit shift registers, two accumulators (each having a K-bit full adder

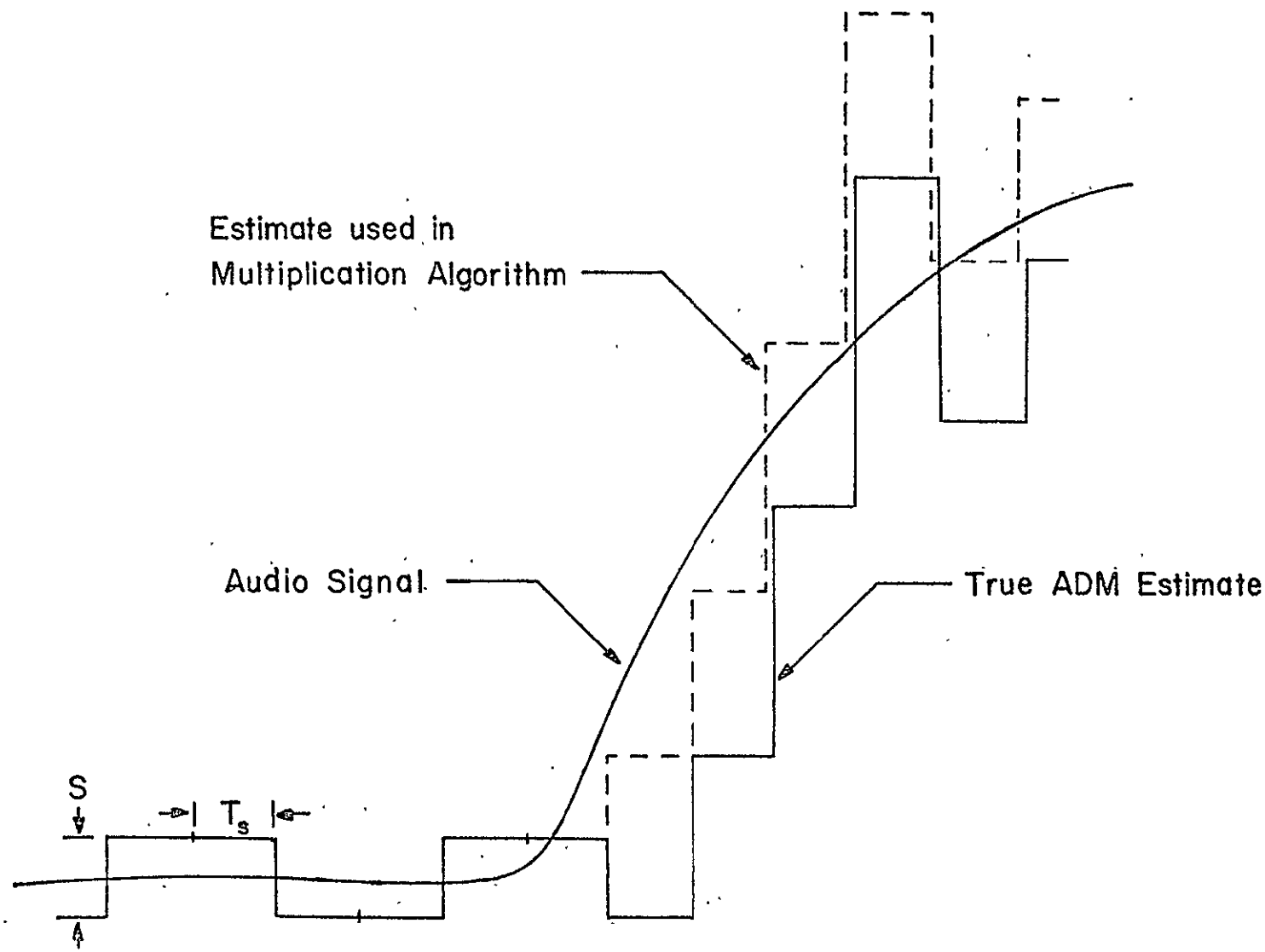


Fig. 2.4-3. Example when the Step Size Relationship is Invalid

for the Song audio mode:

$$S_y(k)\hat{x}(k-1) = S_y(k-1)x(k-2)e_y(k-1)e_y(k-2)$$
$$+ S_x(k-1)S_y(k-1)e_y(k-1)e_y(k-2) \qquad (2.4\text{-}5)$$
$$+ \hat{x}(k-1)Se_y(k-2),$$

$$S_x(k)\hat{y}(k-1) = S_x(k-1)\hat{y}(k-2)e_x(k-1)e_x(k-2)$$
$$+ S_y(k-1)S_x(k-1)e_x(k-1)e_x(k-2) \qquad (2.4\text{-}6)$$
$$+ \hat{y}(k-1)Se_x(k-2),$$

$$S_x(k)S_y(k) = |S_x(k-1)S_y(k-1)|e_x(k-1)e_y(k-1)$$
$$+ S|S_x(k-1)|e_x(k-1)e_y(k-2)$$
$$+ S|S_y(k-1)|e_y(k-1)e_x(k-2) \qquad (2.4\text{-}7)$$
$$+ S^2e_x(k-2)e_y(k-2).$$

Equations (2.4-5), (2.4-6) and (2.4-7) are readily realizable with standard digital hardware similar to the DM adder shown in Fig. 2.2-3. These three terms can be constructed with nothing more complicated than adders, delays, hard-wired scalers and exclusive-OR gates to multiply by ±1 and produce the absolute value.

All of the design structures that we have derived are accumulator type systems. For both the adder (Sec. 2.3) and the multiplier (Sec. 2.4), for all DM modes, the present out-

racy of the product, while, at the same time, increasing the complexity of the hardware.

2.6 SNR with Constant Inputs

Consider first the SNR obtained by adding two statistically independent constant signals, such as x and y, that are PCM encoded. Defining the quantized samples as x_q and y_q, and the respective errors as ε_x and ε_y, the PCM sum is seen to be

$$a_q = x_q + y_q, \tag{2.6-1}$$

where

$$x_q = x - \varepsilon_x \tag{2.6-2}$$

and

$$y_q = y - \varepsilon_y. \tag{2.6-3}$$

We can also express the PCM sum as

$$a_q = a - \varepsilon_a, \tag{2.6-4}$$

where the true sum is

$$a = x + y \tag{2.6-5}$$

and the sum error is

$$\varepsilon_a = \varepsilon_x + \varepsilon_y. \tag{2.6-6}$$

If the minimum step size, S, is the distance between PCM levels and it is sufficiently small, then ε_x and ε_y will be equally likely in the interval [-S/2, S/2]. One can readily show [16] that

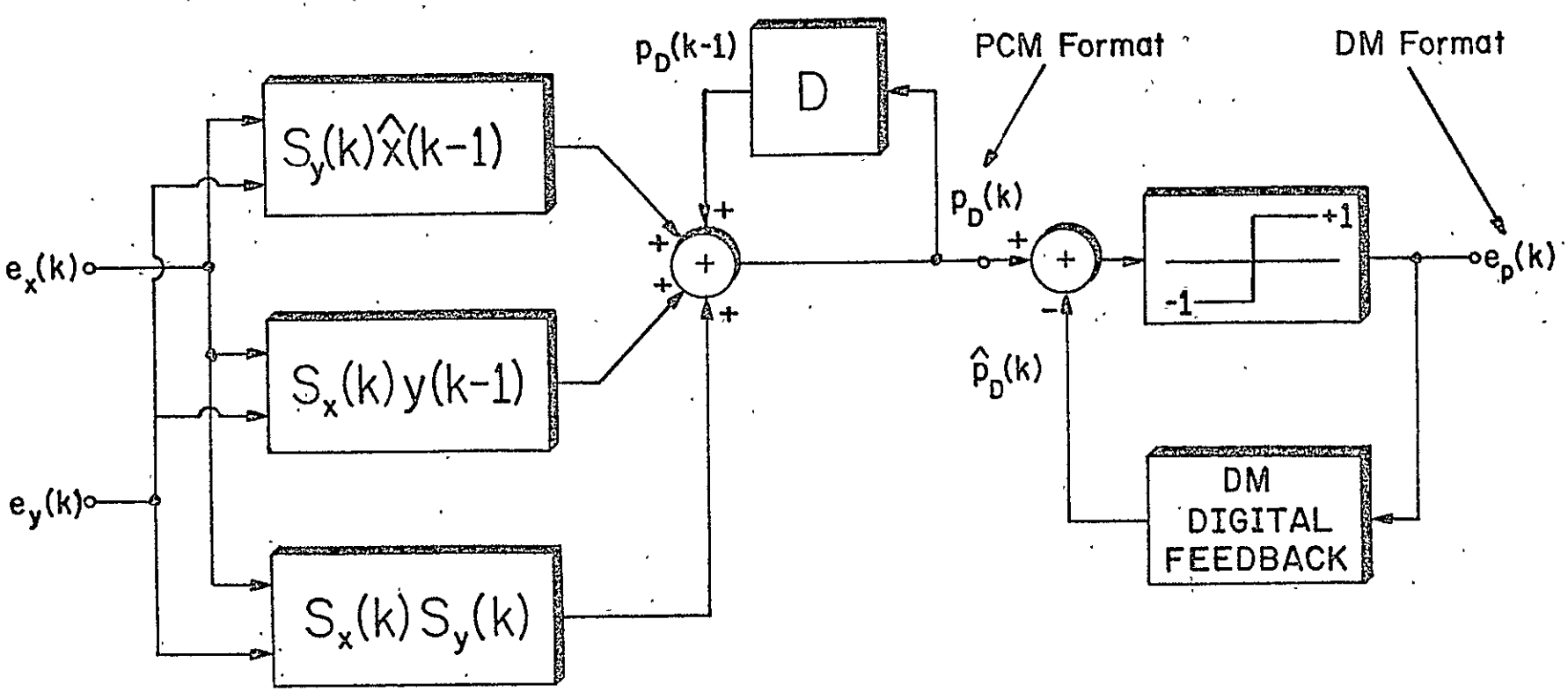


Fig. 2.4-1. Direct Product of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

For the linear DM, no difficulty arises and the direct product is

$$p_D(k) = p_D(k-1) + Se_y(k-1)\hat{x}(k-1) + Se_x(k-1)\hat{y}(k-1) + S^2 e_x(k-1)e_y(k-1). \tag{2.4-3}$$

The realization of this system, shown in Fig. (2.4-2), is extremely easy because there are no non-linear operations, only simple scaling including multiplication by +1 or -1.

To derive the recursive relationships for the partial products with the Song audio mode algorithm, we must use a step size relationship common to all types of DMs, that is,

$$S_x(k) = |S_x(k)|e_x(k-1). \tag{2.4-4}$$

This equation says that the sign of the present step size, $S_x(k)$, is dictated by the past DM output bit, $e_x(k-1)$. From Eq. (2.1-6), we see that this property is applicable for the Song audio mode as long as $|S_x(k-1)| \geq S$. If $S_x(k-1) = 0$ and $e_x(k-1) = e_x(k-2)$, then this property is also valid. Only when $S_x(k-1) = 0$ and $e_x(k-1) \neq e_x(k-2)$, the step size relationship becomes invalid. This invalidity is caused by a hardware limitation that allows the step size to be zero rather than an arbitrarily small value. We shall show, however, that the condition of invalidity has a very low probability of occurrence and the resulting signal estimate used in the multiplication algorithm does not substantially degrade the product.

sign presented does, however, perform well and this will be seen from the simulation responses for elementary input waveforms and also from the SNR performance curves. There are other design techniques that could have been incorporated into the direct product design. We could introduce a "leak" factor in Eq. (2.4-2) and feedback a fraction of the output, $p_D(k)$, to generate, say $0.9p_D(k - 1)$. Alternately, we might use a different averaging filter after $p_D(k)$; or we could perform some kind of averaging on $\hat{x}(k)$ and $\hat{y}(k)$ before we form their product. Each of these ideas would have to be analyzed individually to determine its merits and shortcomings in formulating the direct DM product.

2.5 Hardware Complexity

From Eq. (2.2-2) or Fig. 2.2-1 we see that the complexity and quantity of the hardware needed for a DM adder is essentially equivalent to that needed for a PCM adder. In PCM addition, since we have two K-bit words coming from $x(t)$ and $y(t)$, we require two K-bit input storage registers, one K-bit full adder and, since we do not allow overflow, one K-bit output register. To obtain the DM direct sum, we need the step size circuitry for both signals (some of which can be time shared) terminating in registers with less than K-bit capacity, one transfer register with enough bits to represent twice the maximum step size, one K-bit full adder and one K-bit delay register.

The comparison of hardware complexity for multiplica-

tion is somewhat more involved. PCM multiplication is generally treated as a static operation where two K-bit words are either fed into a combinatorial circuit, or into a preprogrammed ROM, or into a repeating add-store-and-shift circuit. We must also remember that to multiply two signals, bandlimited to f_m, we must perform this static operation on the PCM words from the two signals at a rate of $4f_m$ since the product will be bandlimited to $2f_m$.

Now we can examine the hardware complexity needed for these PCM multipliers. A combinatorial circuit needs two K-bit input storage registers, K^2 AND gates, K K-bit full adders and a 2K-bit output storage register. This is easily seen by observing the structure that arises when we use "long" multiplication to obtain the product of two K-bit words, $A_K \cdots A_2 A_1$ and $B_K \cdots B_2 B_1$. A ROM, with a 2K-bit input address, normally has 2^{2K} memory locations. Even though there are only K^2 different product values, the ROM must still use 2^{2K} memory locations to multiply as well as needing input and output registers. The repeating add-store-and-shift device requires two K-bit input registers, a K-bit shift register, K AND gates, a 2K-bit full adder and a 2K-bit output register. However, for this last multiplier, we must perform K repeated additions in the time interval $1/4f_m$ before we obtain the final product word.

Considering the DM multiplier for the linear mode, as shown in Fig. 2.4-2, the hardware needed is two 2-bit shift registers, two accumulators (each having a K-bit full adder

sign presented does, however, perform well and this will be seen from the simulation responses for elementary input waveforms and also from the SNR performance curves. There are other design techniques that could have been incorporated into the direct product design. We could introduce a "leak" factor in Eq. (2.4-2) and feedback a fraction of the output, $p_D(k)$, to generate, say $0.9p_D(k - 1)$. Alternately, we might use a different averaging filter after $p_D(k)$; or we could perform some kind of averaging on $\hat{x}(k)$ and $\hat{y}(k)$ before we form their product. Each of these ideas would have to be analyzed individually to determine its merits and shortcomings in formulating the direct DM product.

2.5 Hardware Complexity

From Eq. (2.2-2) or Fig. 2.2-1 we see that the complexity and quantity of the hardware needed for a DM adder is essentially equivalent to that needed for a PCM adder. In PCM addition, since we have two K-bit words coming from $x(t)$ and $y(t)$, we require two K-bit input storage registers, one K-bit full adder and, since we do not allow overflow, one K-bit output register. To obtain the DM direct sum, we need the step size circuitry for both signals (some of which can be time shared) terminating in registers with less than K-bit capacity, one transfer register with enough bits to represent twice the maximum step size, one K-bit full adder and one K-bit delay register.

The comparison of hardware complexity for multiplica-

tion is somewhat more involved. PCM multiplication is generally treated as a static operation where two K-bit words are either fed into a combinatorial circuit, or into a preprogrammed ROM, or into a repeating add-store-and-shift circuit. We must also remember that to multiply two signals, bandlimited to f_m, we must perform this static operation on the PCM words from the two signals at a rate of $4f_m$ since the product will be bandlimited to $2f_m$.

Now we can examine the hardware complexity needed for these PCM multipliers. A combinatorial circuit needs two K-bit input storage registers, K^2 AND gates, K K-bit full adders and a 2K-bit output storage register. This is easily seen by observing the structure that arises when we use "long" multiplication to obtain the product of two K-bit words, $A_K \cdots A_2 A_1$ and $B_K \cdots B_2 B_1$. A ROM, with a 2K-bit input address, normally has 2^{2K} memory locations. Even though there are only K^2 different product values, the ROM must still use 2^{2K} memory locations to multiply as well as needing input and output registers. The repeating add-store-and-shift device requires two K-bit input registers, a K-bit shift register, K AND gates, a 2K-bit full adder and a 2K-bit output register. However, for this last multiplier, we must perform K repeated additions in the time interval $1/4f_m$ before we obtain the final product word.

Considering the DM multiplier for the linear mode, as shown in Fig. 2.4-2, the hardware needed is two 2-bit shift registers, two accumulators (each having a K-bit full adder

and a K-bit storage register), 2K+1 exclusive-OR gates, a 2K-bit transfer register, a 2K-bit full adder and a 2K-bit output register. This hardware complexity is equivalent to that needed in the repeating add-store-and-shift PCM multiplier. Of course, the amount of hardware needed for the DM device depends on the step size algorithm employed.

Finally, we observe that, as the number of bits increases, the amount of hardware increases linearly with K for any DM multiplier. This is not true for all PCM multipliers. The complexity of the combinatorial circuit and the ROM increases as K^2 and 2^{2K}, respectively. Although the complexity of the repeating add-store-and-shift circuit increases linearly with K, the time required to obtain the product of two K-bit words also increases linearly with K.

When evaluating the total complexity of our DM multiplier, we conclude that the hardware required is comparable to a PCM multiplier. Signals are generally encoded with an ADM, where the step size algorithm, and consequently the required product circuitry, is somewhat more involved than when a linear DM is used as the encoder. However, the DM multiplier performs all its operations directly on the DM bit streams, a concept never considered heretofore in digital signal processing where DM signals were involved. The conventional alternative is to convert the DM bits into PCM format first. This would give rise to a conversion error. Then we can perform PCM multiplication, but this would increase the conversion error and further add to the inaccu-

and a K-bit storage register), 2K+1 exclusive-OR gates, a 2K-bit transfer register, a 2K-bit full adder and a 2K-bit output register. This hardware complexity is equivalent to that needed in the repeating add-store-and-shift PCM multiplier. Of course, the amount of hardware needed for the DM device depends on the step size algorithm employed.

Finally, we observe that, as the number of bits increases, the amount of hardware increases linearly with K for any DM multiplier. This is not true for all PCM multipliers. The complexity of the combinatorial circuit and the ROM increases as K^2 and 2^{2K}, respectively. Although the complexity of the repeating add-store-and-shift circuit increases linearly with K, the time required to obtain the product of two K-bit words also increases linearly with K.

When evaluating the total complexity of our DM multiplier, we conclude that the hardware required is comparable to a PCM multiplier. Signals are generally encoded with an ADM, where the step size algorithm, and consequently the required product circuitry, is somewhat more involved than when a linear DM is used as the encoder. However, the DM multiplier performs all its operations directly on the DM bit streams, a concept never considered heretofore in digital signal processing where DM signals were involved. The conventional alternative is to convert the DM bits into PCM format first. This would give rise to a conversion error. Then we can perform PCM multiplication, but this would increase the conversion error and further add to the inaccu-

performance comparison with PCM. There are other ways to obtain the product in PCM form by using the samples of both signals at a rate of $2f_m$. This matter is further pursued in App. 2.

The problem that we consider here is the formation of the product of $x(t)$ and $y(t)$ when we only have their DM sequences, $\{e_x(k)\}$ and $\{e_y(k)\}$, available. Once again we restrict ourselves to a design structure that can be implemented with standard digital hardware. Forming the direct product as the product of the individual signal estimates, we have

$$p_D(k) = \hat{x}(k)\hat{y}(k). \tag{2.4-1}$$

As in the case of the direct sum, we can develop a recursive relationship as follows:

$$\begin{aligned} p_D(k) = p_D(k-1) &+ S_y(k)\hat{x}(k-1) + S_x(k)\hat{y}(k-1) \\ &+ S_x(k)S_y(k). \end{aligned} \tag{2.4-2}$$

The basic block diagram showing the direct product, in PCM format $[p_D(k)]$ and in DM format $[e_p(k)]$, is given in Fig. (2.4-1). Although the structure for the direct product is universal for any digital DM definable by Eqs. (2.1-1) through (2.1-3), it will be useful only if the step size algorithm is such that we can recursively realize the particle products, that is, the last three terms in Eq. (2.4-2).

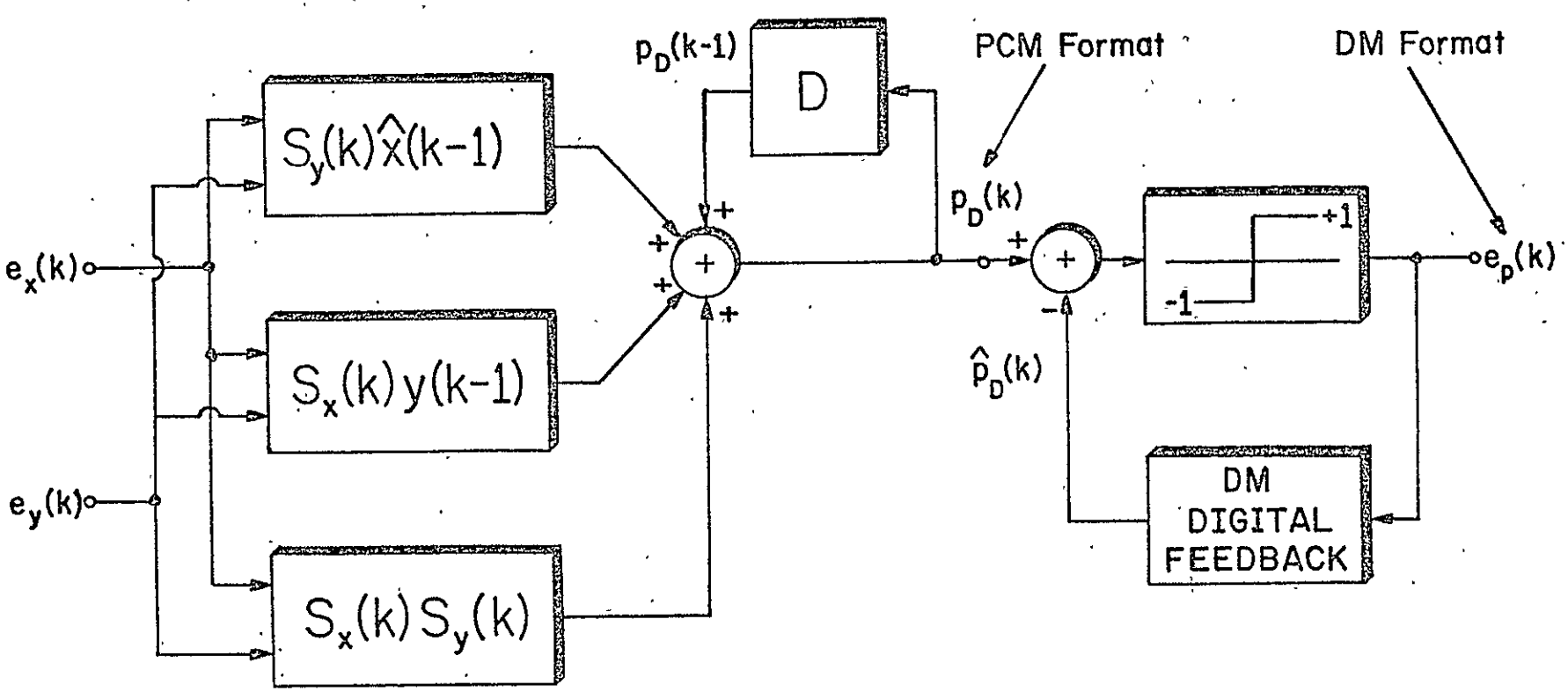


Fig. 2.4-1. Direct Product of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

For the linear DM, no difficulty arises and the direct product is

$$p_D(k) = p_D(k-1) + Se_y(k-1)\hat{x}(k-1) + Se_x(k-1)\hat{y}(k-1) + S^2 e_x(k-1)e_y(k-1). \tag{2.4-3}$$

The realization of this system, shown in Fig. (2.4-2), is extremely easy because there are no non-linear operations, only simple scaling including multiplication by +1 or -1.

To derive the recursive relationships for the partial products with the Song audio mode algorithm, we must use a step size relationship common to all types of DMs, that is,

$$S_x(k) = |S_x(k)|e_x(k-1). \tag{2.4-4}$$

This equation says that the sign of the present step size, $S_x(k)$, is dictated by the past DM output bit, $e_x(k-1)$. From Eq. (2.1-6), we see that this property is applicable for the Song audio mode as long as $|S_x(k-1)| \geq S$. If $S_x(k-1) = 0$ and $e_x(k-1) = e_x(k-2)$, then this property is also valid. Only when $S_x(k-1) = 0$ and $e_x(k-1) \neq e_x(k-2)$, the step size relationship becomes invalid. This invalidity is caused by a hardware limitation that allows the step size to be zero rather than an arbitrarily small value. We shall show, however, that the condition of invalidity has a very low probability of occurrence and the resulting signal estimate used in the multiplication algorithm does not substantially degrade the product.

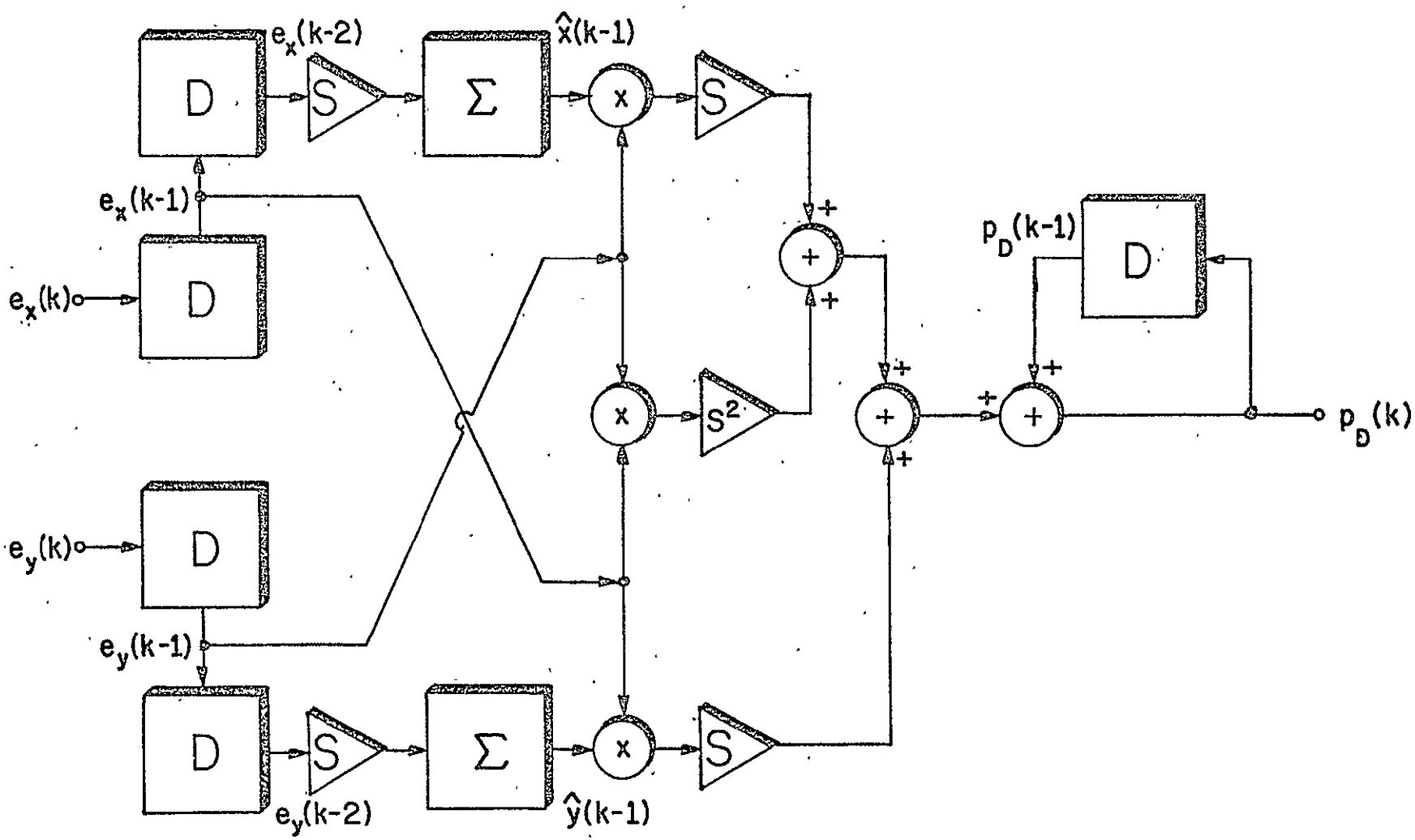


Fig. 2.4-2. DM Multiplier for Linear Mode

Equation (2.4-4) can only be invalid when $S_x(k) = 0$. A zero step size occurs primarily when the ADM is in its minimum steady state pattern. That is, the estimate resembles Fig. 2.3-1 when m = 0. This corresponds to the audio signal being zero because, in speech, 50% of the time there is no voice. Recently, step size statistics have been obtained for the Song audio mode ADM, using actual speech signals. They show that the probability of a zero step size is approximately 0.04 when f_s = 32K bits/second. Since $S_x(k) = 0$ occurs twice in a minimum steady state estimate pattern, the probability of such a pattern is 0.08. Let us assume that the audio signal is equally likely to increase or decrease from its zero value in any of the four periods of the steady state pattern. Only 2 of the 8 signal variations give rise to the condition when Eq. (2.4-4) will be invalid. Therefore, the probability of invalidity is 0.02.

We have created a situation in Fig. 2.4-3 where the step size relationship is invalid. The solid curve represents the true ADM estimate and the broken line waveform is the estimate used in the multiplication algorithm. From this figure, we observe that the estimate used in the multiplication algorithm is just as good an approximation to the audio signal as the true estimate. We shall see, in Sec. 2.8.3, that the use of Eq. (2.4-2) does not noticeably effect the output SNR of the DM multiplier.

Now that we have justified the step size relationship, we can use it to express recursively the partial products

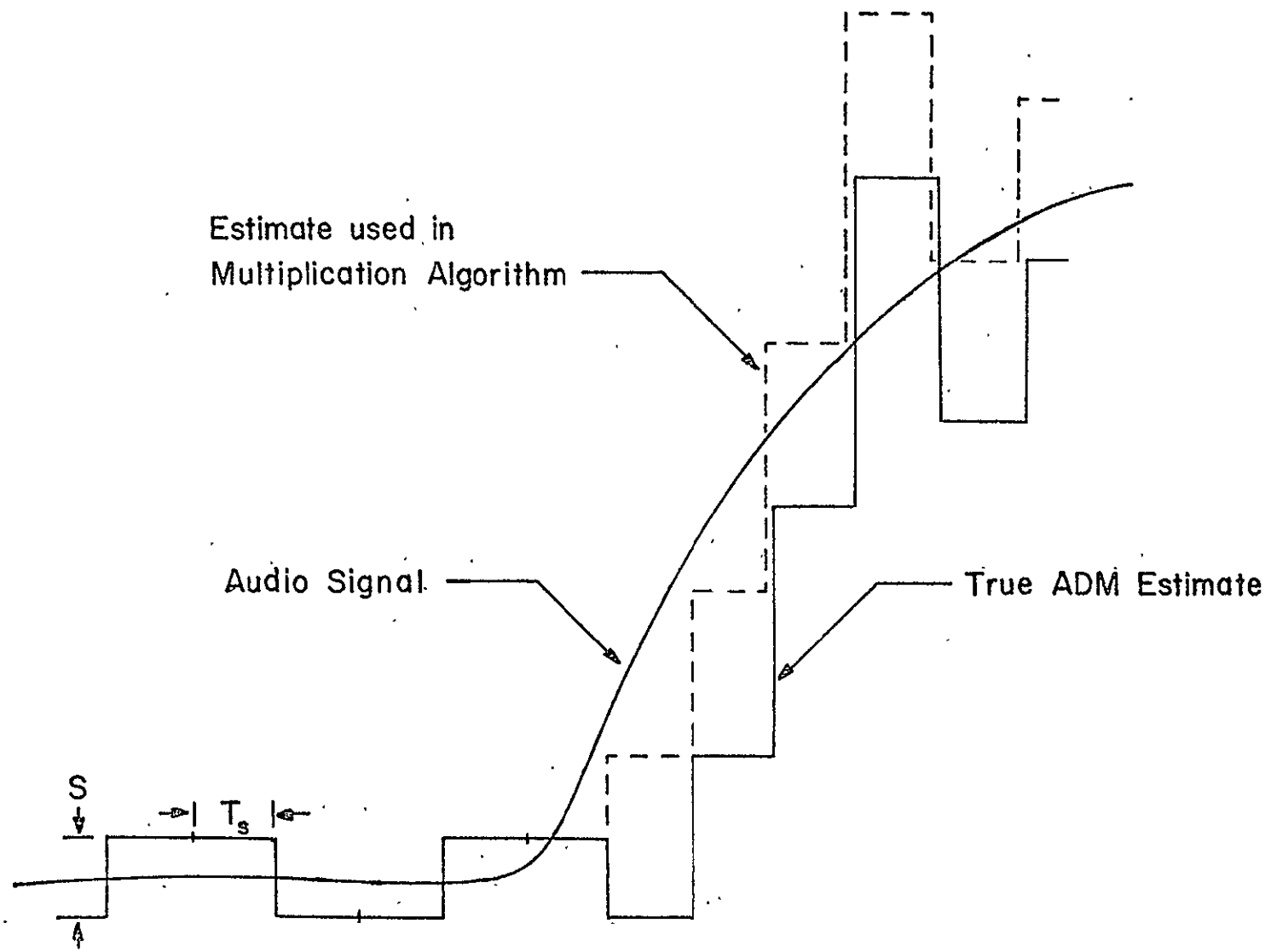


Fig. 2.4-3. Example when the Step Size Relationship is Invalid

for the Song audio mode:

$$S_y(k)\hat{x}(k-1) = S_y(k-1)x(k-2)e_y(k-1)e_y(k-2)$$
$$+ S_x(k-1)S_y(k-1)e_y(k-1)e_y(k-2) \qquad (2.4\text{-}5)$$
$$+ \hat{x}(k-1)Se_y(k-2),$$

$$S_x(k)\hat{y}(k-1) = S_x(k-1)\hat{y}(k-2)e_x(k-1)e_x(k-2)$$
$$+ S_y(k-1)S_x(k-1)e_x(k-1)e_x(k-2) \qquad (2.4\text{-}6)$$
$$+ \hat{y}(k-1)Se_x(k-2),$$

$$S_x(k)S_y(k) = |S_x(k-1)S_y(k-1)|e_x(k-1)e_y(k-1)$$
$$+ S|S_x(k-1)|e_x(k-1)e_y(k-2)$$
$$+ S|S_y(k-1)|e_y(k-1)e_x(k-2) \qquad (2.4\text{-}7)$$
$$+ S^2e_x(k-2)e_y(k-2).$$

Equations (2.4-5), (2.4-6) and (2.4-7) are readily realizable with standard digital hardware similar to the DM adder shown in Fig. 2.2-3. These three terms can be constructed with nothing more complicated than adders, delays, hard-wired scalers and exclusive-OR gates to multiply by ±1 and produce the absolute value.

All of the design structures that we have derived are accumulator type systems. For both the adder (Sec. 2.3) and the multiplier (Sec. 2.4), for all DM modes, the present out-

put is equal to the past output plus additional terms. Thus, it is important to begin with the correct initial condition for the past output, or else suffer a constant offset error. It is convenient to start with both signals, x(t) and y(t), at zero so that we can employ a zero initial condition for the past output.

As in the case of the direct sum, we expect the direct product, since it is formulated as the product of the individual signal estimates, to exhibit a periodic pattern when responding to constant inputs. When using the Song audio mode algorithm, the direct product generates four possible steady state waveforms. In Fig. 2.4-4, we show the general structure of a steady state waveform. The values of C_1 and C_2 depend upon x_q and y_q and the amplitude of the steady state error pattern ($\hat{x} - x_q$ and $\hat{y} - y_q$), while d_1 and d_2 depend only on the latter of these two. The numerical values of d_1 and d_2 can be entirely different, but they both have the same form, as can be seen by multiplying two steady state patterns together, that is,

$$|d_i| = L/4, \tag{2.4-8}$$

where

$$i = 1, 2,$$

$$L = \text{a positive integer}$$

and

$$L \leq 4M^2 + 4M + 1 \tag{2.4-9}$$

where M is bounded in Eq. (2.3-4).

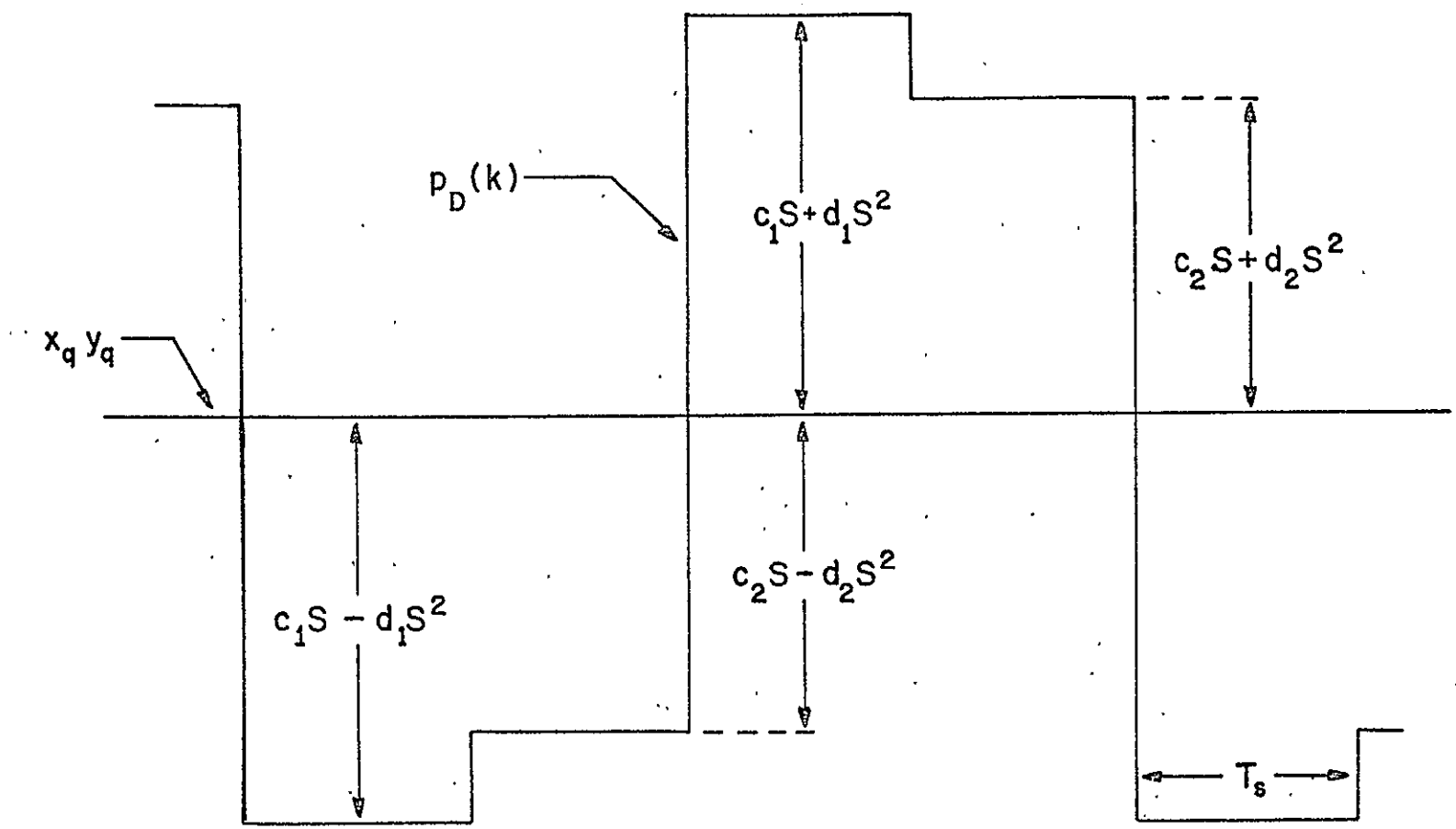


Fig. 2.4-4. Steady State Direct Product for Song Audio Mode DM with Constant Inputs

The arithmetic average of any four consecutive values of $p_D(k)$ always equals the product of the quantized values of the inputs plus a second-order term depending on S^2. This warrants the use of the following four-term non-recursive filter after $p_D(k)$:

$$P(k) = \frac{1}{4}[p_D(k) + p_D(k-1) + p_D(k-2) + p_D(k-3)]. \quad (2.4\text{-}10)$$

Applying Eq. (2.4-10) to the waveform shown in Fig. 2.4-3, we see that

$$P(k) = x_q y_q + (d_1 + d_2)S^2/2 \quad (2.4\text{-}11)$$

for all k, as long as $p_D(k)$ has reached the steady state. We have found, through computer simulations, that the factor $(d_1 + d_2)/2$ generally is no larger than 10 or 20. In a practical DM encoder with 10 bits of internal arithmetic and an amplitude range of $V_{pp} = 10$ volts, the minimum step size, S, will be approximately 10 millivolts. Therefore, the second-order term will be in the order of 1-2 millivolts. Even if we allow $(d_1 + d_2)/2$ to be 100, the error only reaches 10 millivolts or one minimum step size. Certainly, one step size out of 1024 can be considered insignificant. For any reasonably small value of step size, the second order term, $(d_1 + d_2)S^2/2$, is negligible and thus, after the four-term averaging filter, the DM product yields a result almost identical to the PCM product.

The direct DM product design structure that we have derived is not a unique solution to this problem. The de-

sign presented does, however, perform well and this will be seen from the simulation responses for elementary input waveforms and also from the SNR performance curves. There are other design techniques that could have been incorporated into the direct product design. We could introduce a "leak" factor in Eq. (2.4-2) and feedback a fraction of the output, $p_D(k)$, to generate, say $0.9p_D(k - 1)$. Alternately, we might use a different averaging filter after $p_D(k)$; or we could perform some kind of averaging on $\hat{x}(k)$ and $\hat{y}(k)$ before we form their product. Each of these ideas would have to be analyzed individually to determine its merits and shortcomings in formulating the direct DM product.

2.5 Hardware Complexity

From Eq. (2.2-2) or Fig. 2.2-1 we see that the complexity and quantity of the hardware needed for a DM adder is essentially equivalent to that needed for a PCM adder. In PCM addition, since we have two K-bit words coming from $x(t)$ and $y(t)$, we require two K-bit input storage registers, one K-bit full adder and, since we do not allow overflow, one K-bit output register. To obtain the DM direct sum, we need the step size circuitry for both signals (some of which can be time shared) terminating in registers with less than K-bit capacity, one transfer register with enough bits to represent twice the maximum step size, one K-bit full adder and one K-bit delay register.

The comparison of hardware complexity for multiplica-

tion is somewhat more involved. PCM multiplication is generally treated as a static operation where two K-bit words are either fed into a combinatorial circuit, or into a preprogrammed ROM, or into a repeating add-store-and-shift circuit. We must also remember that to multiply two signals, bandlimited to f_m, we must perform this static operation on the PCM words from the two signals at a rate of $4f_m$ since the product will be bandlimited to $2f_m$.

Now we can examine the hardware complexity needed for these PCM multipliers. A combinatorial circuit needs two K-bit input storage registers, K^2 AND gates, K K-bit full adders and a 2K-bit output storage register. This is easily seen by observing the structure that arises when we use "long" multiplication to obtain the product of two K-bit words, $A_K \cdots A_2 A_1$ and $B_K \cdots B_2 B_1$. A ROM, with a 2K-bit input address, normally has 2^{2K} memory locations. Even though there are only K^2 different product values, the ROM must still use 2^{2K} memory locations to multiply as well as needing input and output registers. The repeating add-store-and-shift device requires two K-bit input registers, a K-bit shift register, K AND gates, a 2K-bit full adder and a 2K-bit output register. However, for this last multiplier, we must perform K repeated additions in the time interval $1/4f_m$ before we obtain the final product word.

Considering the DM multiplier for the linear mode, as shown in Fig. 2.4-2, the hardware needed is two 2-bit shift registers, two accumulators (each having a K-bit full adder

and then evaluate the performance. Since the inputs are statistically independent, the product signal power is

$$\overline{p^2} = \sigma_x^2 \sigma_y^2 \ . \tag{2.6-24}$$

For PCM signals, the sample value and the error are likewise statistically independent. Consequently,

$$\overline{\varepsilon}_p = 0 \tag{2.6-25a}$$

and

$$\mathrm{Var}(\varepsilon_p) = \overline{\varepsilon_p^2} = \overline{x^2}\ \overline{\varepsilon_y^2} + \overline{y^2}\ \overline{\varepsilon_x^2} + \overline{\varepsilon_x^2}\ \overline{\varepsilon_y^2} \ . \tag{2.6-25b}$$

Evaluating the variance of the product error, we obtain

$$\mathrm{Var}(\varepsilon_p) = (\sigma_x^2 + \sigma_y^2) S^2/12 + S^4/144 \ . \tag{2.6-26}$$

Thus, for the product of PCM encoded signals,

$$\mathrm{SNR}_p(\mathrm{PCM}) = \frac{144\sigma_x^2\sigma_y^2}{12(\sigma_x^2 + \sigma_y^2)S^2 + S^4} \ . \tag{2.6-27}$$

If the two signals have equal power and the step size is small ($\sigma_x^2 = \sigma_y^2 = \sigma^2 >> S^2$), the SNR becomes

$$\mathrm{SNR}_p(\mathrm{PCM}) \simeq 6\sigma^2/S^2 \ . \tag{2.6-28}$$

For the direct product of DM encoded signals, we can similarly develop an expression for the SNR. Again, we include the averaging filter introduced in Eq. (2.4-10) and the DM error is formulated as

$$\xi_p = p - P \ , \tag{2.6-29}$$

where

$$P = P(k) = x_q y_q + \delta S^2 \ . \qquad (2.6\text{-}30)$$

Here we have assumed that $p_D(k)$ has reached steady state and δ represents the constant, $(d_1 + d_2)/2$, introduced previously. With the aid of Eqs. (2.6-19) and (2.6-20), we find that

$$\xi_p = \varepsilon_p + \delta S^2 \ . \qquad (2.6\text{-}31)$$

Let us define the product SNR for DM signals to be

$$SNR_p(DM) \equiv \frac{\overline{p^2}}{Var(\xi_p)} \ . \qquad (2.6\text{-}32)$$

To calculate $SNR_p(DM)$, we must evaluate

$$\overline{\xi_p} = \delta S^2 \ , \qquad (2.6\text{-}33a)$$

$$\overline{\xi_p^2} = \overline{\varepsilon_p^2} + \delta^2 S^4 \qquad (2.6\text{-}33b)$$

and

$$Var(\xi_p) = \overline{\xi_p^2} - \overline{\xi_p}^2 = \overline{\varepsilon_p^2} \ . \qquad (2.6\text{-}33c)$$

But Eq. (2.6-33c) implies that

$$Var(\xi_p) = Var(\varepsilon_p) \qquad (2.6\text{-}33d)$$

and

$$SNR_p(DM) = SNR_p(PCM) \ . \qquad (2.6\text{-}34)$$

Therefore,

$$SNR_p(DM) \simeq 6\sigma^2/S^2 \ , \qquad (2.6\text{-}35)$$

racy of the product, while, at the same time, increasing the complexity of the hardware.

2.6 SNR with Constant Inputs

Consider first the SNR obtained by adding two statistically independent constant signals, such as x and y, that are PCM encoded. Defining the quantized samples as x_q and y_q, and the respective errors as ε_x and ε_y, the PCM sum is seen to be

$$a_q = x_q + y_q, \tag{2.6-1}$$

where

$$x_q = x - \varepsilon_x \tag{2.6-2}$$

and

$$y_q = y - \varepsilon_y. \tag{2.6-3}$$

We can also express the PCM sum as

$$a_q = a - \varepsilon_a, \tag{2.6-4}$$

where the true sum is

$$a = x + y \tag{2.6-5}$$

and the sum error is

$$\varepsilon_a = \varepsilon_x + \varepsilon_y. \tag{2.6-6}$$

If the minimum step size, S, is the distance between PCM levels and it is sufficiently small, then ε_x and ε_y will be equally likely in the interval [-S/2, S/2]. One can readily show [16] that

$$\overline{\varepsilon_x} = \overline{\varepsilon_y} = 0 \tag{2.6-7a}$$

and

$$\overline{\varepsilon_x^2} = \overline{\varepsilon_y^2} = S^2/12, \tag{2.6-7b}$$

where the bar in the above equations denotes the probabalistic expected value.

Let us now define the sum SNR for PCM signals to be

$$SNR_a(PCM) \equiv \frac{\overline{a^2}}{Var(\varepsilon_a)}, \tag{2.6-8}$$

where

$$Var(\alpha) \equiv \overline{(\alpha - \overline{\alpha})^2} = \overline{\alpha^2} - \overline{\alpha}^2 . \tag{2.6-9}$$

Assuming x and y to be independent, zero mean, Gaussian random variables with variances σ_x^2 and σ_y^2, the variance of the sum is

$$\overline{a^2} = \sigma_x^2 + \sigma_y^2 . \tag{2.6-10}$$

To determine the variance of the sum error, $Var(\varepsilon_a)$, we note that

$$\overline{\varepsilon_a} = 0 , \tag{2.6-11a}$$

so that

$$Var(\varepsilon_a) = \overline{\varepsilon_a^2} = \overline{\varepsilon_x^2} + \overline{\varepsilon_y^2} = S^2/6 . \tag{2.6-11b}$$

Thus, for PCM encoded signals,

$$SNR_a(PCM) = 6(\sigma_x^2 + \sigma_y^2)/S^2 . \tag{2.6-12a}$$

Setting $\sigma_x^2 = \sigma_y^2 = \sigma^2$, Eq. (2.6-12a) becomes

$$SNR_a(PCM) = 12\sigma^2/S^2 \ . \qquad (2.6\text{-}12b)$$

For the direct sum of DM encoded signals, an analysis similar to the PCM case can be conducted by evaluating the difference between a and a_D. However, the actual error that we are concerned with is after the non-recursive four-term averaging filter introduced in Eq. (2.3-6). This error is

$$\xi_A = a - A \ , \qquad (2.6\text{-}13)$$

where

$$A = A(k) = x_q + y_q \ , \qquad (2.6\text{-}14)$$

as obtained in Eq. (2.3-7) when $a_D(k)$ has reached steady state. With the help of Eqs. (2.6-1) and (2.6-4), we see that

$$\xi_A = \varepsilon_a \ , \qquad (2.6\text{-}15)$$

that is, the DM sum error after the averaging filter is exactly the same as the PCM sum error.

Defining the sum SNR for DM signals to be

$$SNR_a(DM) \equiv \frac{\overline{a^2}}{Var(\xi_A)} \ , \qquad (2.6\text{-}16)$$

it is apparent that

$$SNR_a(DM) = SNR_a(PCM) \ . \qquad (2.6\text{-}17)$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Consequently,

$$SNR_a(DM) = 12\sigma^2/S^2 \ , \qquad (2.6\text{-}18)$$

where we have assumed that both x and y possess the same signal power, σ^2. The conclusion is that, for the case of constant input signals, the performance of the direct DM adder followed by the averaging filter is identical to the PCM adder performance when the comparison is based on SNR.

Now we shall consider the SNR obtained by multiplying two constant, statistically independent signals that are PCM encoded. Using the same notation as in the sum analysis, the PCM product is

$$p_q = x_q y_q \ . \qquad (2.6\text{-}19)$$

Expressing the PCM product as

$$p_q = p - \varepsilon_p \ , \qquad (2.6\text{-}20)$$

and using the defining relationships for x_q and y_q in Eq. (2.6-19), we see that the true product is

$$p = xy \qquad (2.6\text{-}21)$$

and the product error is

$$\varepsilon_p = x\varepsilon_y + y\varepsilon_x - \varepsilon_x\varepsilon_y . \qquad (2.6\text{-}22)$$

We can specify the product SNR for PCM signals as

$$SNR_p(PCM) \equiv \frac{\overline{p^2}}{Var(\varepsilon_p)} \ , \qquad (2.6\text{-}23)$$

and then evaluate the performance. Since the inputs are statistically independent, the product signal power is

$$\overline{p^2} = \sigma_x^2 \sigma_y^2 \ . \tag{2.6-24}$$

For PCM signals, the sample value and the error are likewise statistically independent. Consequently,

$$\overline{\varepsilon}_p = 0 \tag{2.6-25a}$$

and

$$\mathrm{Var}(\varepsilon_p) = \overline{\varepsilon_p^2} = \overline{x^2}\ \overline{\varepsilon_y^2} + \overline{y^2}\ \overline{\varepsilon_x^2} + \overline{\varepsilon_x^2}\ \overline{\varepsilon_y^2} \ . \tag{2.6-25b}$$

Evaluating the variance of the product error, we obtain

$$\mathrm{Var}(\varepsilon_p) = (\sigma_x^2 + \sigma_y^2) S^2/12 + S^4/144 \ . \tag{2.6-26}$$

Thus, for the product of PCM encoded signals,

$$\mathrm{SNR}_p(\mathrm{PCM}) = \frac{144\sigma_x^2\sigma_y^2}{12(\sigma_x^2 + \sigma_y^2)S^2 + S^4} \ . \tag{2.6-27}$$

If the two signals have equal power and the step size is small ($\sigma_x^2 = \sigma_y^2 = \sigma^2 >> S^2$), the SNR becomes

$$\mathrm{SNR}_p(\mathrm{PCM}) \simeq 6\sigma^2/S^2 \ . \tag{2.6-28}$$

For the direct product of DM encoded signals, we can similarly develop an expression for the SNR. Again, we include the averaging filter introduced in Eq. (2.4-10) and the DM error is formulated as

$$\xi_p = p - P \ , \tag{2.6-29}$$

performance comparison with PCM. There are other ways to obtain the product in PCM form by using the samples of both signals at a rate of $2f_m$. This matter is further pursued in App. 2.

The problem that we consider here is the formation of the product of $x(t)$ and $y(t)$ when we only have their DM sequences, $\{e_x(k)\}$ and $\{e_y(k)\}$, available. Once again we restrict ourselves to a design structure that can be implemented with standard digital hardware. Forming the direct product as the product of the individual signal estimates, we have

$$p_D(k) = \hat{x}(k)\hat{y}(k). \tag{2.4-1}$$

As in the case of the direct sum, we can develop a recursive relationship as follows:

$$\begin{aligned} p_D(k) = p_D(k-1) &+ S_y(k)\hat{x}(k-1) + S_x(k)\hat{y}(k-1) \\ &+ S_x(k)S_y(k). \end{aligned} \tag{2.4-2}$$

The basic block diagram showing the direct product, in PCM format $[p_D(k)]$ and in DM format $[e_p(k)]$, is given in Fig. (2.4-1). Although the structure for the direct product is universal for any digital DM definable by Eqs. (2.1-1) through (2.1-3), it will be useful only if the step size algorithm is such that we can recursively realize the particle products, that is, the last three terms in Eq. (2.4-2).

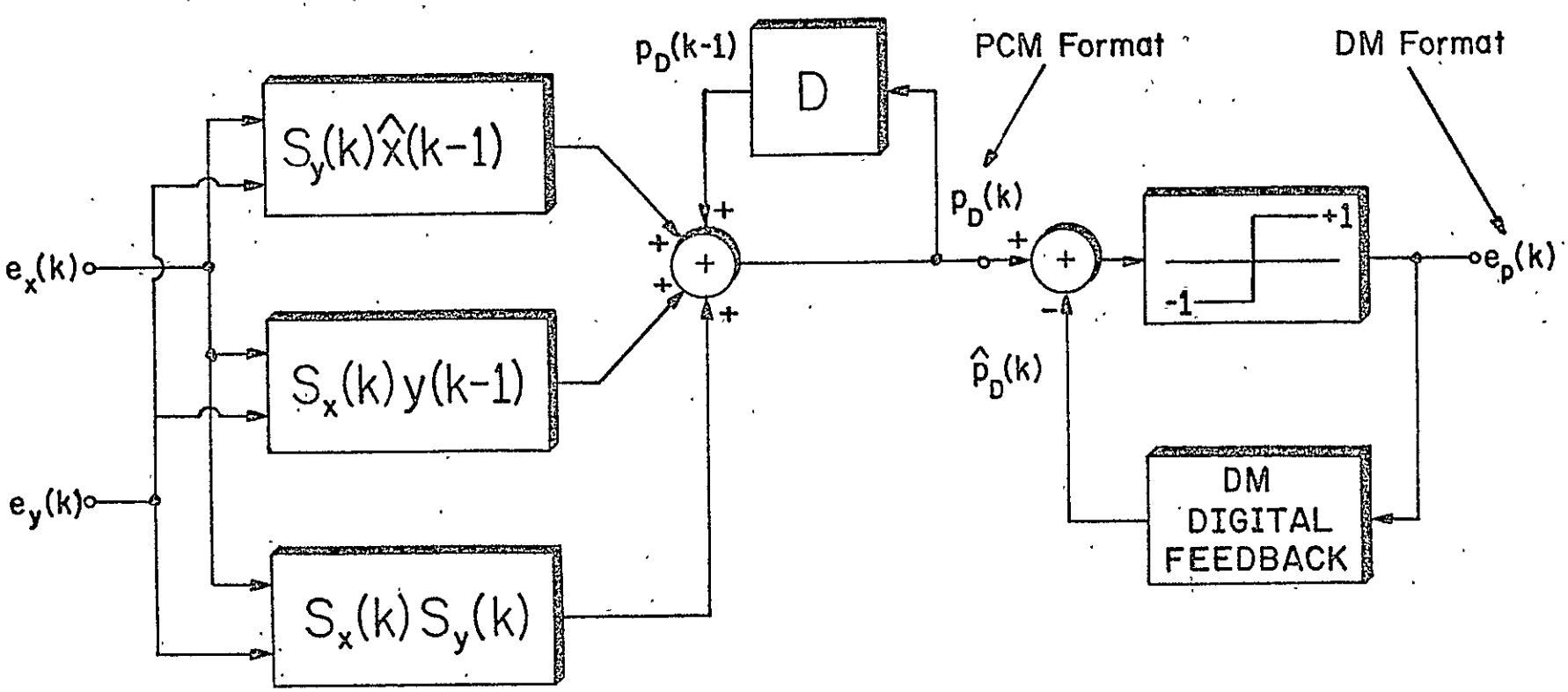


Fig. 2.4-1. Direct Product of DM Encoded Signals

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

For the linear DM, no difficulty arises and the direct product is

$$p_D(k) = p_D(k-1) + Se_y(k-1)\hat{x}(k-1) + Se_x(k-1)\hat{y}(k-1) + S^2 e_x(k-1)e_y(k-1). \tag{2.4-3}$$

The realization of this system, shown in Fig. (2.4-2), is extremely easy because there are no non-linear operations, only simple scaling including multiplication by +1 or -1.

To derive the recursive relationships for the partial products with the Song audio mode algorithm, we must use a step size relationship common to all types of DMs, that is,

$$S_x(k) = |S_x(k)|e_x(k-1). \tag{2.4-4}$$

This equation says that the sign of the present step size, $S_x(k)$, is dictated by the past DM output bit, $e_x(k-1)$. From Eq. (2.1-6), we see that this property is applicable for the Song audio mode as long as $|S_x(k-1)| \geq S$. If $S_x(k-1) = 0$ and $e_x(k-1) = e_x(k-2)$, then this property is also valid. Only when $S_x(k-1) = 0$ and $e_x(k-1) \neq e_x(k-2)$, the step size relationship becomes invalid. This invalidity is caused by a hardware limitation that allows the step size to be zero rather than an arbitrarily small value. We shall show, however, that the condition of invalidity has a very low probability of occurrence and the resulting signal estimate used in the multiplication algorithm does not substantially degrade the product.

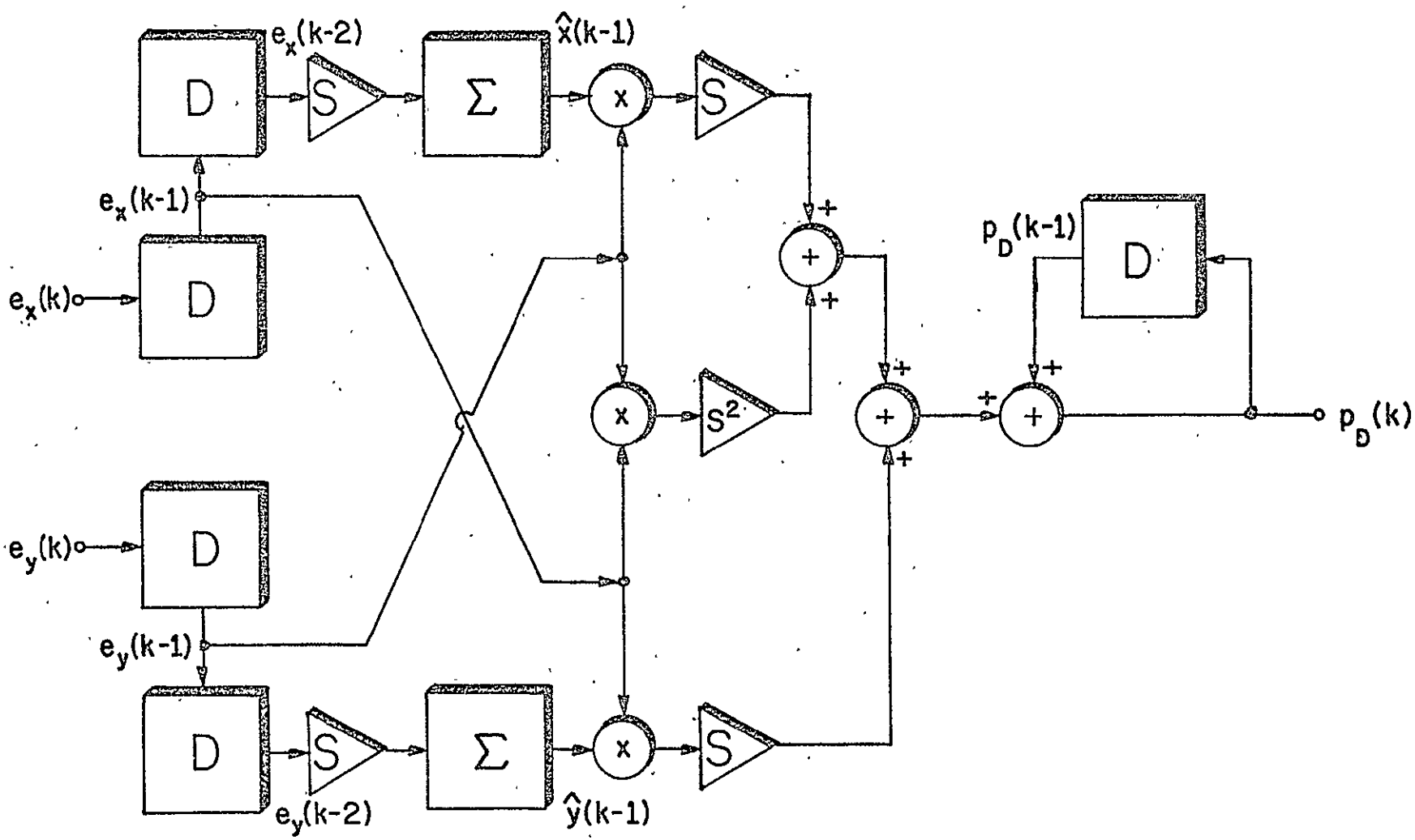


Fig. 2.4-2. DM Multiplier for Linear Mode

Equation (2.4-4) can only be invalid when $S_x(k) = 0$. A zero step size occurs primarily when the ADM is in its minimum steady state pattern. That is, the estimate resembles Fig. 2.3-1 when m = 0. This corresponds to the audio signal being zero because, in speech, 50% of the time there is no voice. Recently, step size statistics have been obtained for the Song audio mode ADM, using actual speech signals. They show that the probability of a zero step size is approximately 0.04 when f_s = 32K bits/second. Since $S_x(k) = 0$ occurs twice in a minimum steady state estimate pattern, the probability of such a pattern is 0.08. Let us assume that the audio signal is equally likely to increase or decrease from its zero value in any of the four periods of the steady state pattern. Only 2 of the 8 signal variations give rise to the condition when Eq. (2.4-4) will be invalid. Therefore, the probability of invalidity is 0.02.

We have created a situation in Fig. 2.4-3 where the step size relationship is invalid. The solid curve represents the true ADM estimate and the broken line waveform is the estimate used in the multiplication algorithm. From this figure, we observe that the estimate used in the multiplication algorithm is just as good an approximation to the audio signal as the true estimate. We shall see, in Sec. 2.8.3, that the use of Eq. (2.4-2) does not noticeably effect the output SNR of the DM multiplier.

Now that we have justified the step size relationship, we can use it to express recursively the partial products

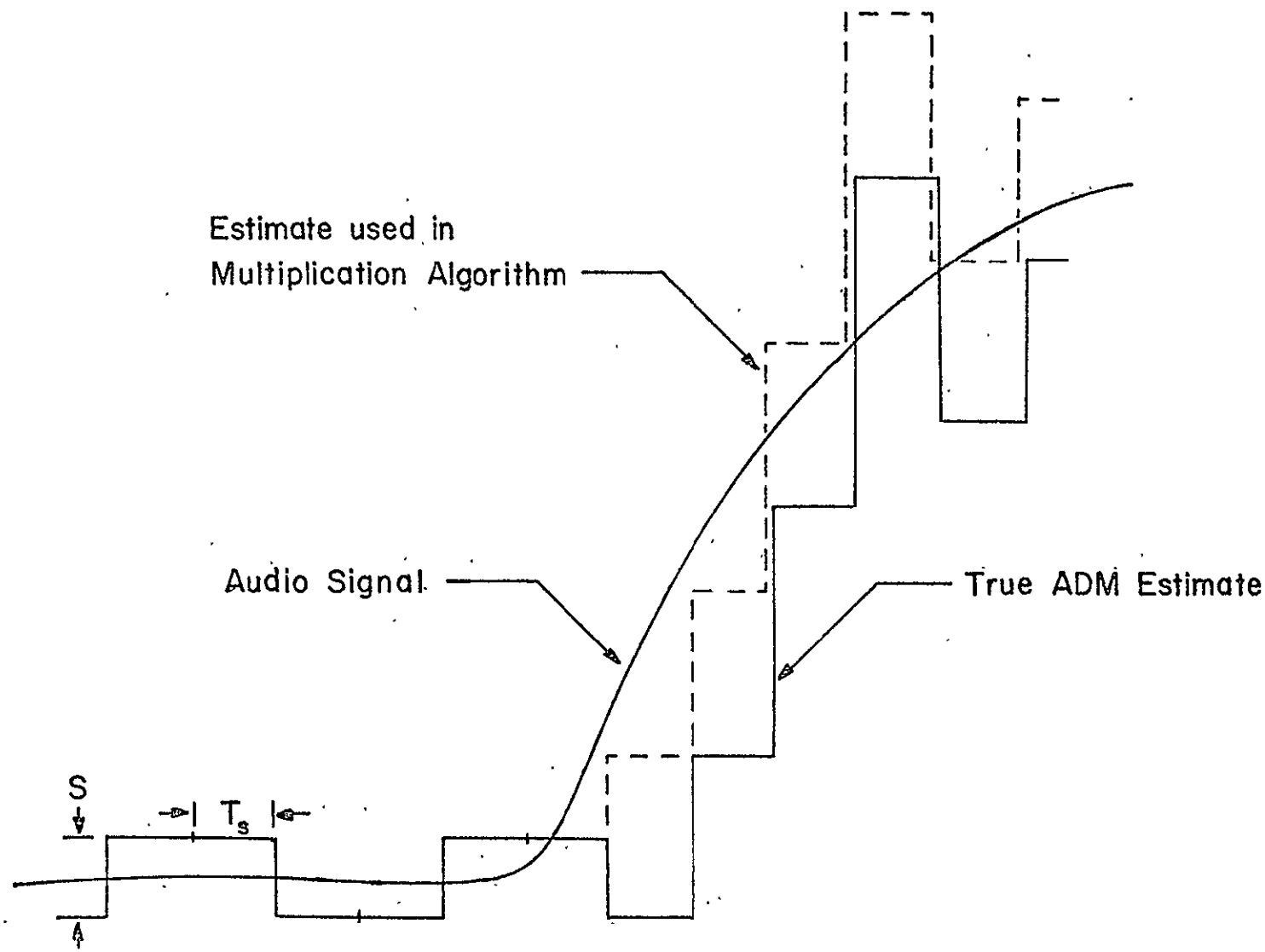


Fig. 2.4-3. Example when the Step Size Relationship is Invalid

for the Song audio mode:

$$S_y(k)\hat{x}(k-1) = S_y(k-1)x(k-2)e_y(k-1)e_y(k-2)$$
$$+ S_x(k-1)S_y(k-1)e_y(k-1)e_y(k-2) \qquad (2.4\text{-}5)$$
$$+ \hat{x}(k-1)Se_y(k-2),$$

$$S_x(k)\hat{y}(k-1) = S_x(k-1)\hat{y}(k-2)e_x(k-1)e_x(k-2)$$
$$+ S_y(k-1)S_x(k-1)e_x(k-1)e_x(k-2) \qquad (2.4\text{-}6)$$
$$+ \hat{y}(k-1)Se_x(k-2),$$

$$S_x(k)S_y(k) = |S_x(k-1)S_y(k-1)|e_x(k-1)e_y(k-1)$$
$$+ S|S_x(k-1)|e_x(k-1)e_y(k-2)$$
$$+ S|S_y(k-1)|e_y(k-1)e_x(k-2) \qquad (2.4\text{-}7)$$
$$+ S^2e_x(k-2)e_y(k-2).$$

Equations (2.4-5), (2.4-6) and (2.4-7) are readily realizable with standard digital hardware similar to the DM adder shown in Fig. 2.2-3. These three terms can be constructed with nothing more complicated than adders, delays, hard-wired scalers and exclusive-OR gates to multiply by ±1 and produce the absolute value.

All of the design structures that we have derived are accumulator type systems. For both the adder (Sec. 2.3) and the multiplier (Sec. 2.4), for all DM modes, the present out-

put is equal to the past output plus additional terms. Thus, it is important to begin with the correct initial condition for the past output, or else suffer a constant offset error. It is convenient to start with both signals, x(t) and y(t), at zero so that we can employ a zero initial condition for the past output.

As in the case of the direct sum, we expect the direct product, since it is formulated as the product of the individual signal estimates, to exhibit a periodic pattern when responding to constant inputs. When using the Song audio mode algorithm, the direct product generates four possible steady state waveforms. In Fig. 2.4-4, we show the general structure of a steady state waveform. The values of C_1 and C_2 depend upon x_q and y_q and the amplitude of the steady state error pattern ($\hat{x} - x_q$ and $\hat{y} - y_q$), while d_1 and d_2 depend only on the latter of these two. The numerical values of d_1 and d_2 can be entirely different, but they both have the same form, as can be seen by multiplying two steady state patterns together, that is,

$$|d_i| = L/4, \tag{2.4-8}$$

where

$$i = 1, 2,$$

$$L = \text{a positive integer}$$

and

$$L \leq 4M^2 + 4M + 1 \tag{2.4-9}$$

where M is bounded in Eq. (2.3-4).

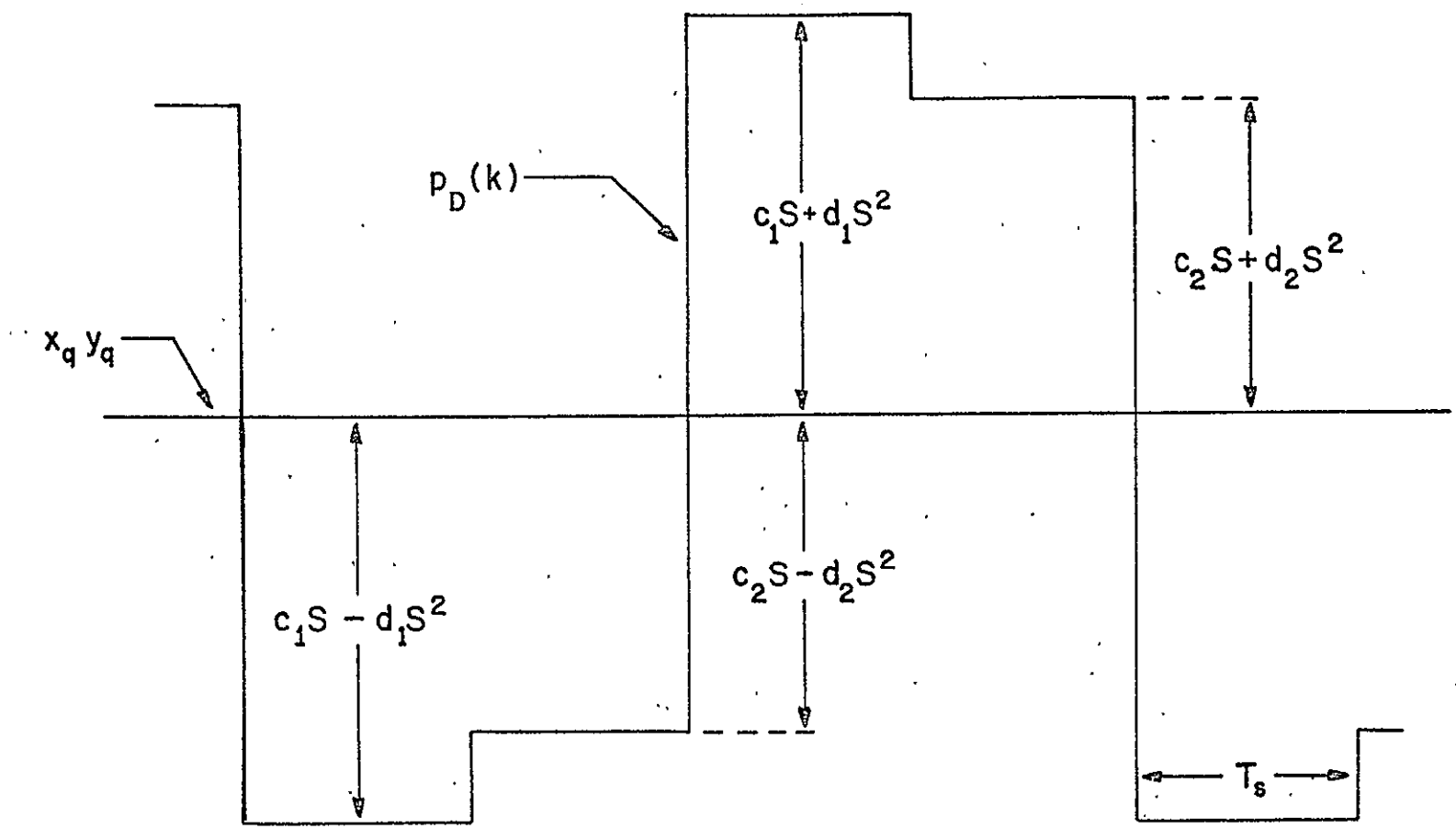


Fig. 2.4-4. Steady State Direct Product for Song Audio Mode DM with Constant Inputs

The arithmetic average of any four consecutive values of $p_D(k)$ always equals the product of the quantized values of the inputs plus a second-order term depending on S^2. This warrants the use of the following four-term non-recursive filter after $p_D(k)$:

$$P(k) = \frac{1}{4}[p_D(k) + p_D(k-1) + p_D(k-2) + p_D(k-3)]. \quad (2.4\text{-}10)$$

Applying Eq. (2.4-10) to the waveform shown in Fig. 2.4-3, we see that

$$P(k) = x_q y_q + (d_1 + d_2)S^2/2 \quad (2.4\text{-}11)$$

for all k, as long as $p_D(k)$ has reached the steady state. We have found, through computer simulations, that the factor $(d_1 + d_2)/2$ generally is no larger than 10 or 20. In a practical DM encoder with 10 bits of internal arithmetic and an amplitude range of $V_{pp} = 10$ volts, the minimum step size, S, will be approximately 10 millivolts. Therefore, the second-order term will be in the order of 1-2 millivolts. Even if we allow $(d_1 + d_2)/2$ to be 100, the error only reaches 10 millivolts or one minimum step size. Certainly, one step size out of 1024 can be considered insignificant. For any reasonably small value of step size, the second order term, $(d_1 + d_2)S^2/2$, is negligible and thus, after the four-term averaging filter, the DM product yields a result almost identical to the PCM product.

The direct DM product design structure that we have derived is not a unique solution to this problem. The de-

sign presented does, however, perform well and this will be seen from the simulation responses for elementary input waveforms and also from the SNR performance curves. There are other design techniques that could have been incorporated into the direct product design. We could introduce a "leak" factor in Eq. (2.4-2) and feedback a fraction of the output, $p_D(k)$, to generate, say $0.9p_D(k - 1)$. Alternately, we might use a different averaging filter after $p_D(k)$; or we could perform some kind of averaging on $\hat{x}(k)$ and $\hat{y}(k)$ before we form their product. Each of these ideas would have to be analyzed individually to determine its merits and shortcomings in formulating the direct DM product.

2.5 Hardware Complexity

From Eq. (2.2-2) or Fig. 2.2-1 we see that the complexity and quantity of the hardware needed for a DM adder is essentially equivalent to that needed for a PCM adder. In PCM addition, since we have two K-bit words coming from $x(t)$ and $y(t)$, we require two K-bit input storage registers, one K-bit full adder and, since we do not allow overflow, one K-bit output register. To obtain the DM direct sum, we need the step size circuitry for both signals (some of which can be time shared) terminating in registers with less than K-bit capacity, one transfer register with enough bits to represent twice the maximum step size, one K-bit full adder and one K-bit delay register.

The comparison of hardware complexity for multiplica-

tion is somewhat more involved. PCM multiplication is generally treated as a static operation where two K-bit words are either fed into a combinatorial circuit, or into a preprogrammed ROM, or into a repeating add-store-and-shift circuit. We must also remember that to multiply two signals, bandlimited to f_m, we must perform this static operation on the PCM words from the two signals at a rate of $4f_m$ since the product will be bandlimited to $2f_m$.

Now we can examine the hardware complexity needed for these PCM multipliers. A combinatorial circuit needs two K-bit input storage registers, K^2 AND gates, K K-bit full adders and a 2K-bit output storage register. This is easily seen by observing the structure that arises when we use "long" multiplication to obtain the product of two K-bit words, $A_K \cdots A_2 A_1$ and $B_K \cdots B_2 B_1$. A ROM, with a 2K-bit input address, normally has 2^{2K} memory locations. Even though there are only K^2 different product values, the ROM must still use 2^{2K} memory locations to multiply as well as needing input and output registers. The repeating add-store-and-shift device requires two K-bit input registers, a K-bit shift register, K AND gates, a 2K-bit full adder and a 2K-bit output register. However, for this last multiplier, we must perform K repeated additions in the time interval $1/4f_m$ before we obtain the final product word.

Considering the DM multiplier for the linear mode, as shown in Fig. 2.4-2, the hardware needed is two 2-bit shift registers, two accumulators (each having a K-bit full adder

and then evaluate the performance. Since the inputs are statistically independent, the product signal power is

$$\overline{p^2} = \sigma_x^2 \sigma_y^2 \ . \tag{2.6-24}$$

For PCM signals, the sample value and the error are likewise statistically independent. Consequently,

$$\overline{\varepsilon}_p = 0 \tag{2.6-25a}$$

and

$$\mathrm{Var}(\varepsilon_p) = \overline{\varepsilon_p^2} = \overline{x^2}\ \overline{\varepsilon_y^2} + \overline{y^2}\ \overline{\varepsilon_x^2} + \overline{\varepsilon_x^2}\ \overline{\varepsilon_y^2} \ . \tag{2.6-25b}$$

Evaluating the variance of the product error, we obtain

$$\mathrm{Var}(\varepsilon_p) = (\sigma_x^2 + \sigma_y^2) S^2/12 + S^4/144 \ . \tag{2.6-26}$$

Thus, for the product of PCM encoded signals,

$$\mathrm{SNR}_p(\mathrm{PCM}) = \frac{144\sigma_x^2\sigma_y^2}{12(\sigma_x^2 + \sigma_y^2)S^2 + S^4} \ . \tag{2.6-27}$$

If the two signals have equal power and the step size is small ($\sigma_x^2 = \sigma_y^2 = \sigma^2 >> S^2$), the SNR becomes

$$\mathrm{SNR}_p(\mathrm{PCM}) \simeq 6\sigma^2/S^2 \ . \tag{2.6-28}$$

For the direct product of DM encoded signals, we can similarly develop an expression for the SNR. Again, we include the averaging filter introduced in Eq. (2.4-10) and the DM error is formulated as

$$\xi_p = p - P \ , \tag{2.6-29}$$

where

$$P = P(k) = x_q y_q + \delta S^2 \ . \qquad (2.6\text{-}30)$$

Here we have assumed that $p_D(k)$ has reached steady state and δ represents the constant, $(d_1 + d_2)/2$, introduced previously. With the aid of Eqs. (2.6-19) and (2.6-20), we find that

$$\xi_p = \varepsilon_p + \delta S^2 \ . \qquad (2.6\text{-}31)$$

Let us define the product SNR for DM signals to be

$$SNR_p(DM) \equiv \frac{\overline{p^2}}{Var(\xi_p)} \ . \qquad (2.6\text{-}32)$$

To calculate $SNR_p(DM)$, we must evaluate

$$\overline{\xi_p} = \delta S^2 \ , \qquad (2.6\text{-}33a)$$

$$\overline{\xi_p^2} = \overline{\varepsilon_p^2} + \delta^2 S^4 \qquad (2.6\text{-}33b)$$

and

$$Var(\xi_p) = \overline{\xi_p^2} - \overline{\xi_p}^2 = \overline{\varepsilon_p^2} \ . \qquad (2.6\text{-}33c)$$

But Eq. (2.6-33c) implies that

$$Var(\xi_p) = Var(\varepsilon_p) \qquad (2.6\text{-}33d)$$

and

$$SNR_p(DM) = SNR_p(PCM) \ . \qquad (2.6\text{-}34)$$

Therefore,

$$SNR_p(DM) \simeq 6\sigma^2/S^2 \ , \qquad (2.6\text{-}35)$$

where

$$P = P(k) = x_q y_q + \delta S^2 \ . \qquad (2.6\text{-}30)$$

Here we have assumed that $p_D(k)$ has reached steady state and δ represents the constant, $(d_1 + d_2)/2$, introduced previously. With the aid of Eqs. (2.6-19) and (2.6-20), we find that

$$\xi_p = \varepsilon_p + \delta S^2 \ . \qquad (2.6\text{-}31)$$

Let us define the product SNR for DM signals to be

$$SNR_p(DM) \equiv \frac{\overline{p^2}}{Var(\xi_p)} \ . \qquad (2.6\text{-}32)$$

To calculate $SNR_p(DM)$, we must evaluate

$$\overline{\xi_p} = \delta S^2 \ , \qquad (2.6\text{-}33a)$$

$$\overline{\xi_p^2} = \overline{\varepsilon_p^2} + \delta^2 S^4 \qquad (2.6\text{-}33b)$$

and

$$Var(\xi_p) = \overline{\xi_p^2} - \overline{\xi_p}^2 = \overline{\varepsilon_p^2} \ . \qquad (2.6\text{-}33c)$$

But Eq. (2.6-33c) implies that

$$Var(\xi_p) = Var(\varepsilon_p) \qquad (2.6\text{-}33d)$$

and

$$SNR_p(DM) = SNR_p(PCM) \ . \qquad (2.6\text{-}34)$$

Therefore,

$$SNR_p(DM) \simeq 6\sigma^2/S^2 \ , \qquad (2.6\text{-}35)$$

where we have again allowed the two signals to have equal power (σ^2) and taken $\sigma^2 >> S^2$. As in the case of addition, we conclude that, based upon SNR, the performance of the direct DM multiplier followed by the averaging filter is identical to the PCM multiplier performance when dealing with constant input signals.

2.7 Simulation Results with Elementary Signals

The direct arithmetic processors that have been designed exhibit some very encouraging characteristics. They are readily physically realizable without an excessive amount of hardware complexity and, with the averaging filter, the SNRs for constant inputs is the same as obtained with PCM processors. To determine the response to a set of elementary input signals, we simulated the direct DM adder and multiplier on a digital computer. In these simulations, we used the Song audio mode algorithm and thus had to realize the circuit shown in Fig. 2.2-3 for the adder. The DM multiplier was constructed from Eqs. (2.4-2), (2.4-5), (2.4-6) and (2.4-7). We also had to simulate two DM encoders in order to generate the bit streams $\{e_x(k)\}$ and $\{e_y(k)\}$.

All the simulations were performed on a PDP 8/L computer with 8K bytes of memory. First, the DM encoders were constructed and it was confirmed that the estimate varied according to the Song audio mode algorithm, i.e., Eq. (2.1-6). Likewise, it was verified that the steady state

estimate to a constant input took the form shown in Fig. 2.3-1. Then we simulated the system for the direct sum followed by the averaging filter defined by Eq. (2.3-6). Initial tests were performed on the direct adder with step inputs, pulse inputs and sinusoidal signals.

In all simulation results, the step size and the sampling period are normalized to unity. Figure 2.7-1 shows the sum of a step function and a pulse; Fig. 2.7-2 displays the result of adding a step function and a sinusoid; and Fig. 2.7-3 gives the addition of two in-phase sinusoids with the same amplitudes and frequencies. In all cases we have included the actual sum, shifted to account for the processor's delay. With the actual sum, we can visually evaluate these initial tests.

Next, the direct DM product system, again followed by the four-term averaging filter, was successfully simulated. We confirmed that the steady state direct product with constant inputs did, in fact, take the form shown in Fig. 2.4-3. As in the case of the direct sum, we used the same set of elementary signals in our initial tests. In Fig. 2.7-4, we show the product of a step function and a pulse. In Fig. 2.7-5, we display the result of multiplying a step function and a sinusoid. In Fig. 2.7-6, we give the multiplication of two in-phase sinusoids with the same amplitudes and frequencies. Again, we include the actual product delayed to facilitate the evaluation of the direct DM product.

All of these simulation results verify the theory de-

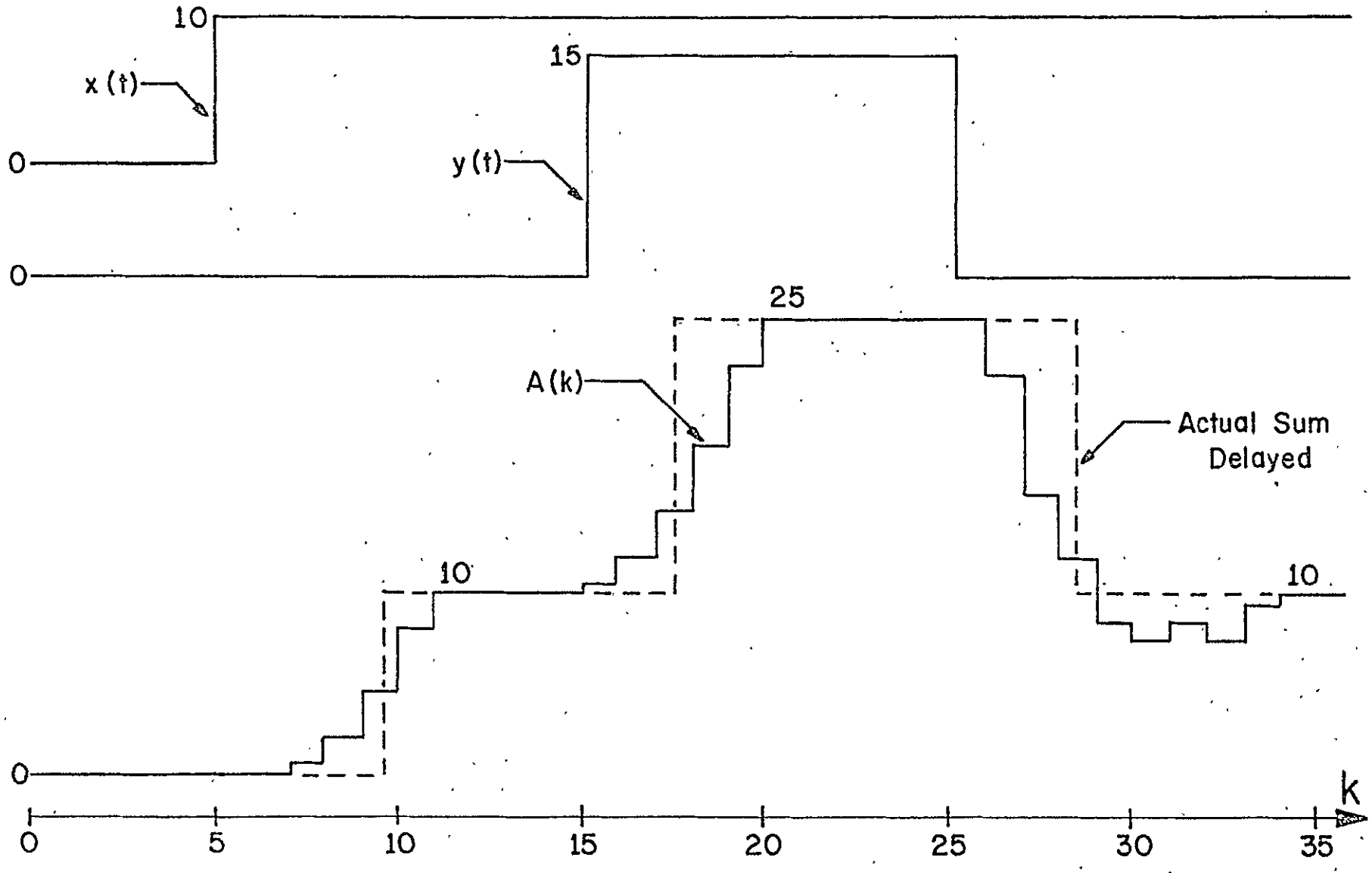


Fig. 2.7-1. DM Sum of Step and Pulse

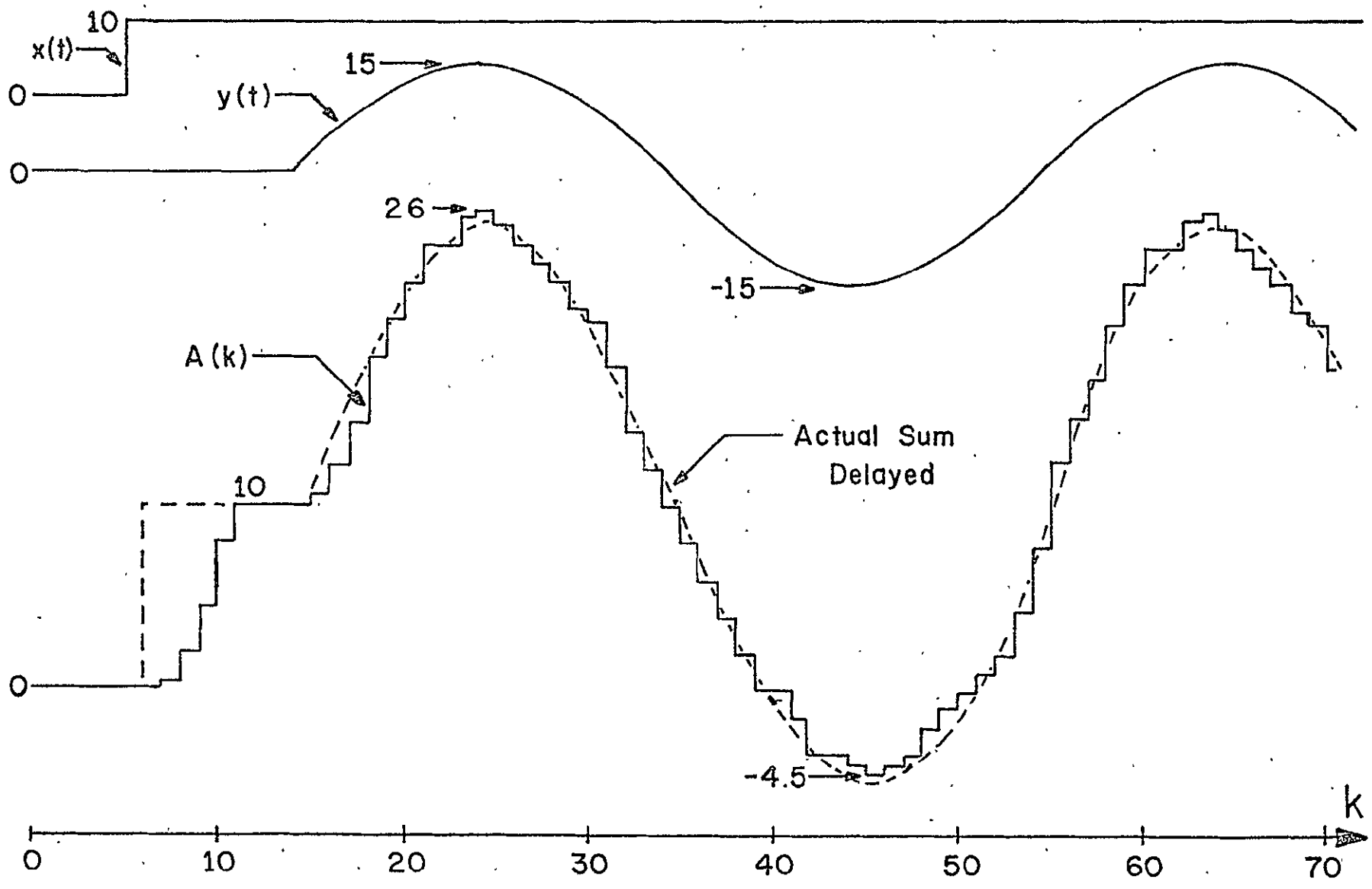


Fig. 2.7-2. DM Sum of a Step and a Sinusoid

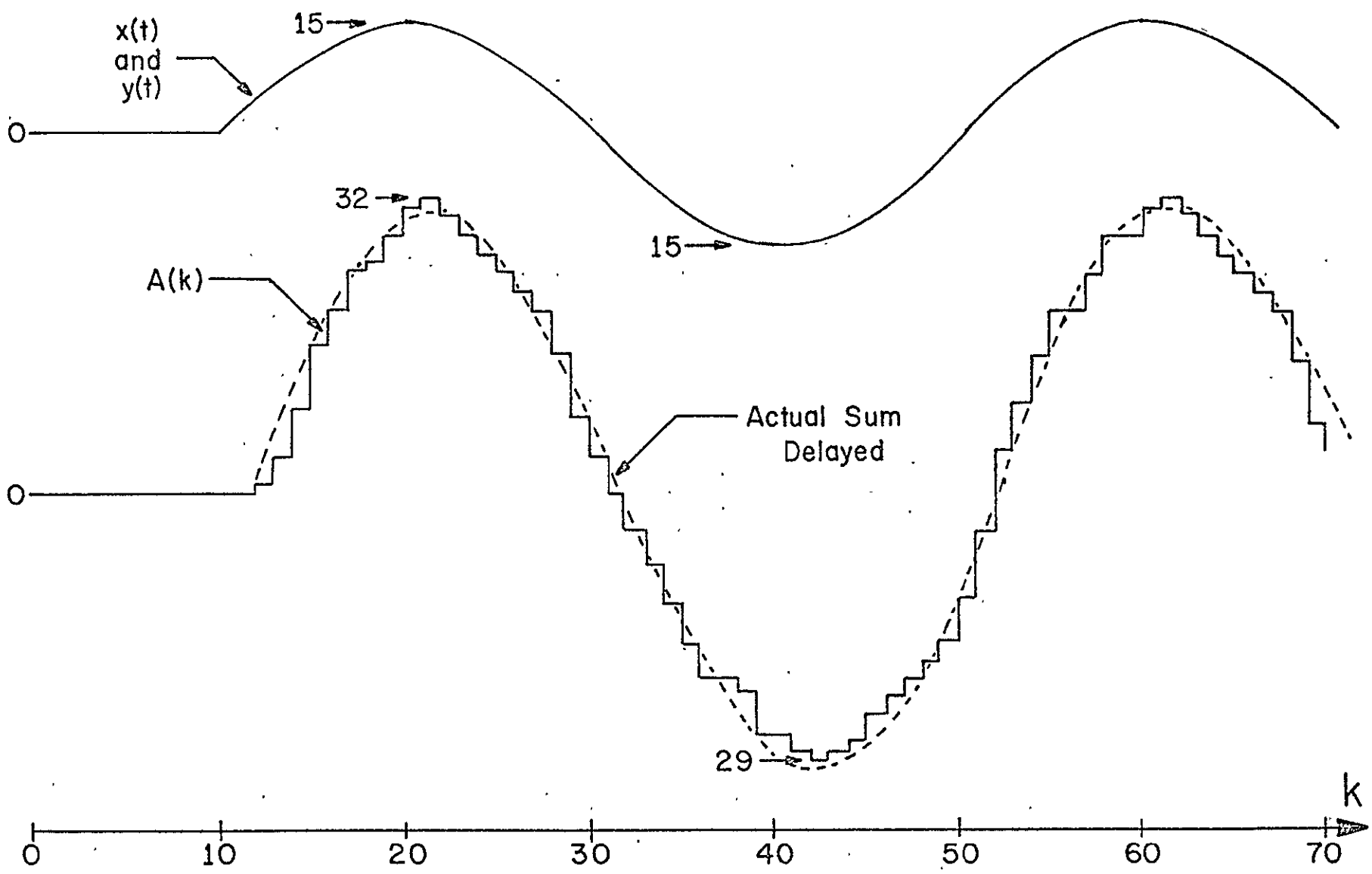


Fig. 2.7-3. DM Sum of Two Sinusoids

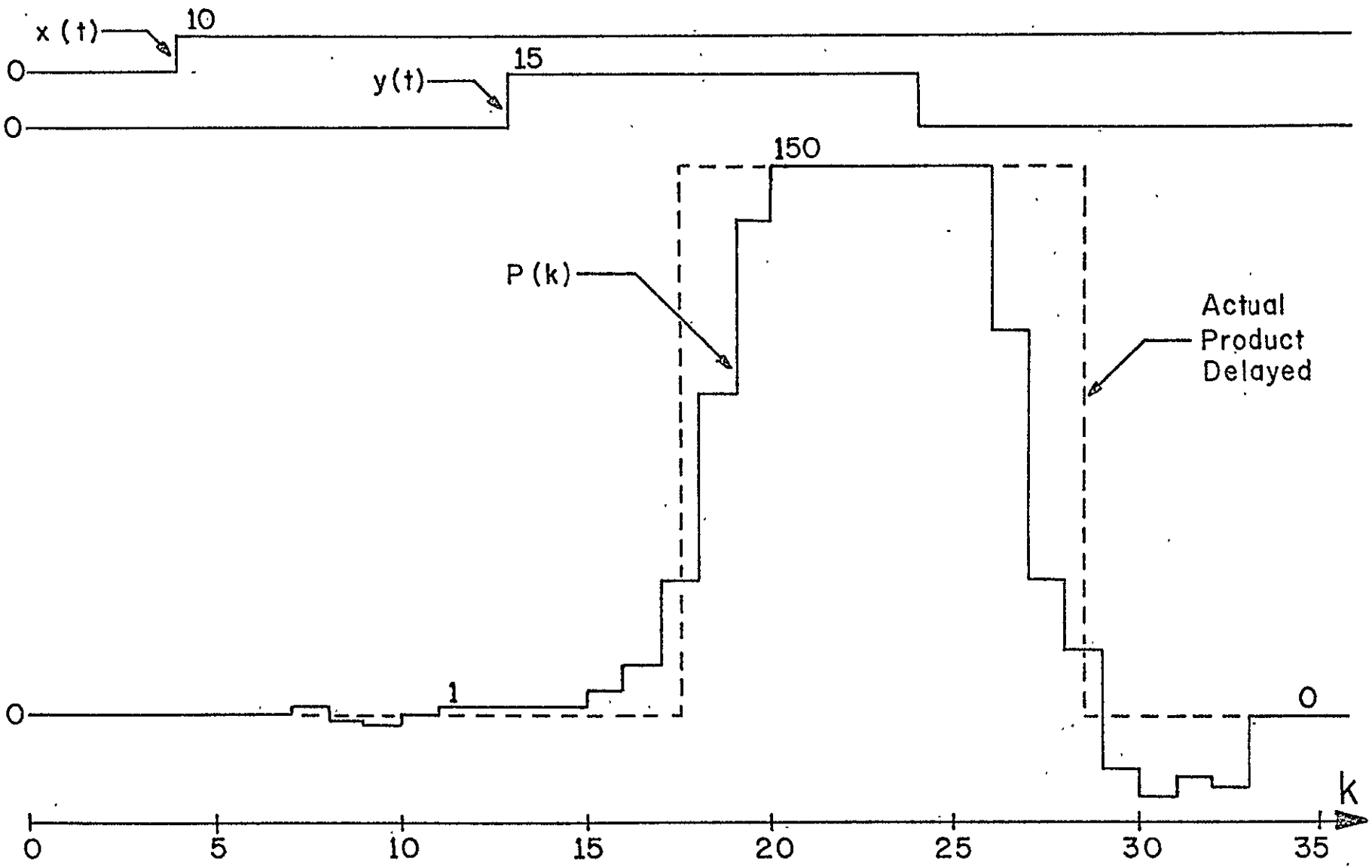


Fig. 2.7-4. DM Product of a Step and a Pulse

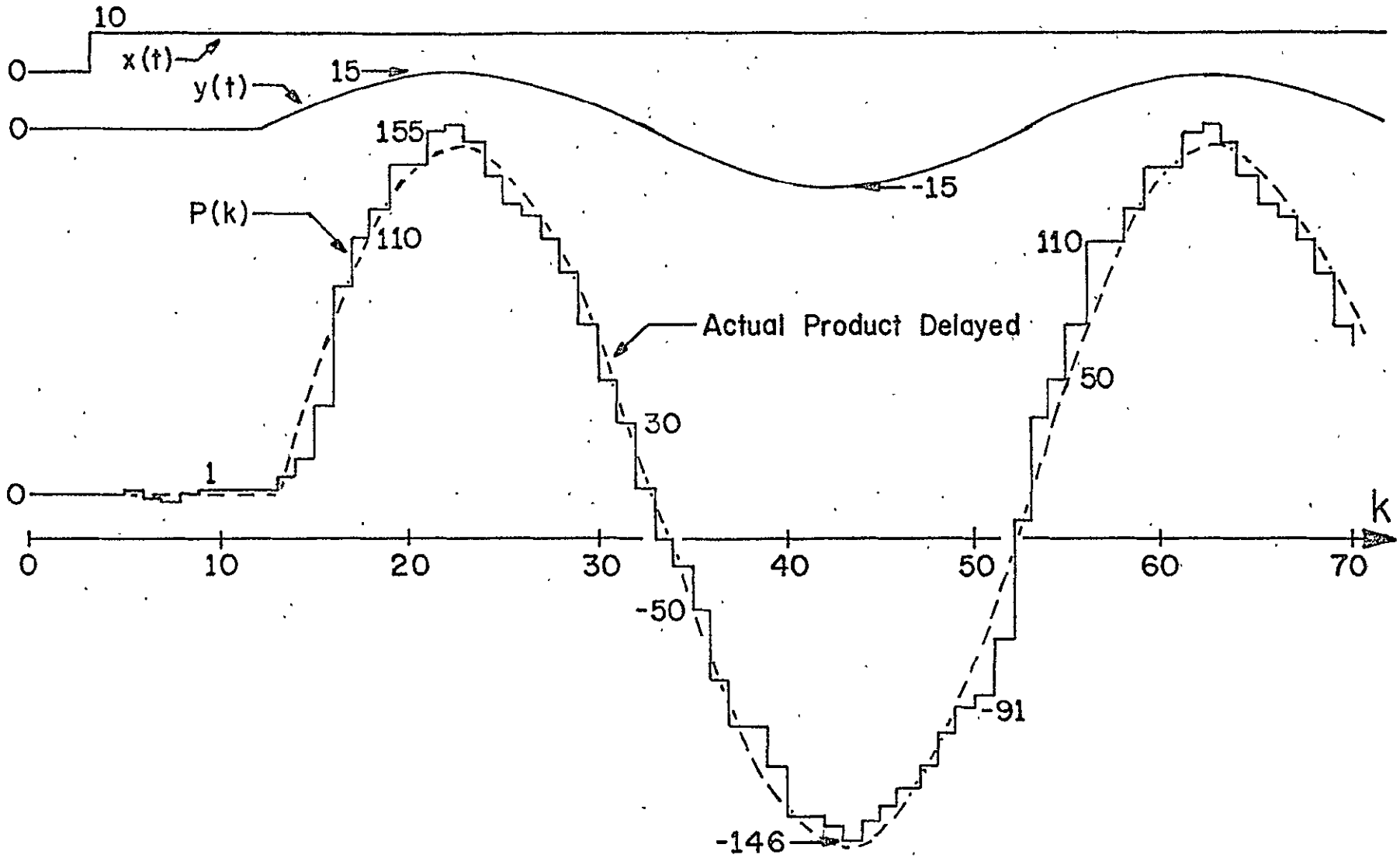


Fig. 2.7-5. DM Product of a Step and a Sinusoid

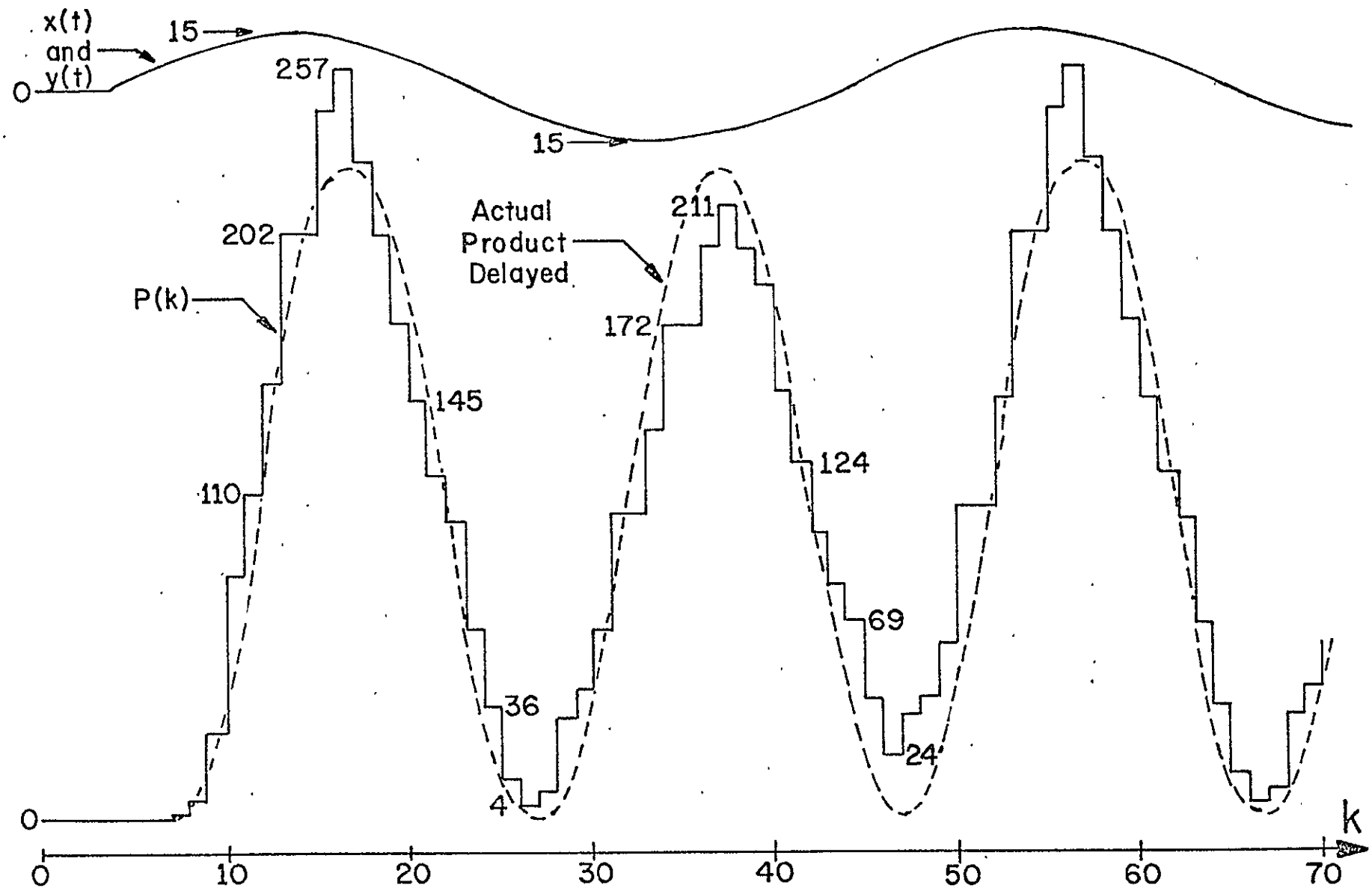


Fig. 2.7-6. DM Product of Two Sinusoids

veloped for both the DM direct sum and the DM direct product. We must emphasize the role played by the four-term averaging filter to achieve both sum and product results that are so accurate. To fully appreciate the effect of this four-term averaging, we show in Fig. 2.7-7 the direct product of a step and a pulse without the four-term averaging. Comparing this with Fig. 2.7-4, which is the result after four-term averaging, clearly demonstrates the important role played by this filter.

As a concluding remark, we observe an important characteristic of the DM multiplier. In Fig. 2.7-8, we show the response of a Song audio mode DM to a step of amplitude 150. The response time, needed to reach 150, is at least 17 sampling periods. Notice, from Fig. 2.7-4, that for the multiplier to reach an amplitude of 150 it takes only 8 sampling periods. Thus, we have expanded the bandwidth by a factor of two, consistent with the previous assumption of the multiplication process.

2.8 Performance Evaluation

The simulation results presented above offer a sufficiently good qualitative evaluation of the operation of the DM adder and multiplier. However, we also would like to obtain a quantitative figure of merit which will allow easy comparison with other digital processing systems. Since we have confined ourselves to a DM audio mode algorithm, we shall apply an evaluation criterion commonly used for

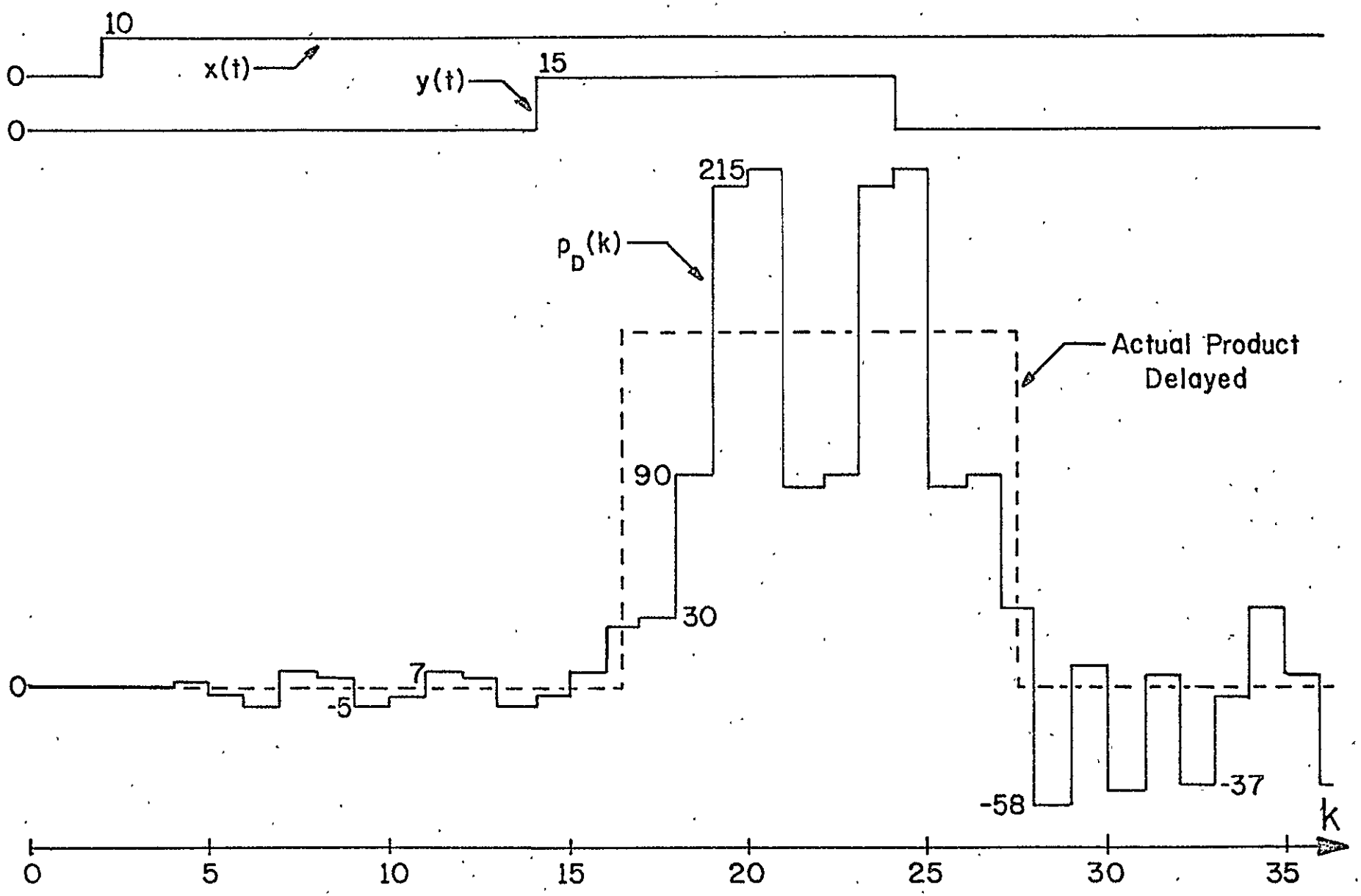


Fig. 2.7-7. DM Product of a Step and a Pulse without Four-Term Averaging Filter

speech waveforms.

Voice signals generally occupy a bandwidth of approximately 2500-3000 Hz, starting about 200-300 Hz and having most of their energy in the area of 600-800 Hz. A voice system is often tested by using a single tone of frequency 600-800 Hz as the input and measuring the output SNR after a LPF which cutsoff at about four times the tone frequency. We have developed a technique to simulate this type of evaluation test on a digital computer and have generated a family of SNR curves for both the DM adder and the DM multiplier.

2.8.1 Fourier Series Representation of the DM Estimate

First, we shall develop the theory needed to calculate the output SNR for the simple case of a DM encoder. Then we shall show that it will be a natural extension to apply this theory to the DM processors.

Let $x(t)$, the input to a DM encoder, be a sinusoid of frequency f_o and let the DM bit rate be

$$f_s = Pf_o, \tag{2.8-1}$$

where P is a positive integer greater than one. Whenever the sampling frequency is an integer multiple of the sinusoid frequency, the DM estimate, $\hat{x}(k)$, will assume a periodic sinusoidal steady state pattern which can be expressed as a Fourier series. The frequency components of this series can be calculated and low pass filtered. The resulting

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

filtered estimate then can be used to determine the output SNR.

A much stronger statement of this concept can be made by not limiting x(t) to be a sinusoid, but only to be a periodic signal. If f_s is now an integral multiple of the fundamental frequency of the input, then we can always determine the spectral composition of the estimate. For our purposes, however, we need only be concerned with sinusoidal inputs.

Consequently, the period of the estimate will be T_o, where

$$T_o = 1/f_o \ , \tag{2.8-2}$$

or any integral multiple of T_o. For the sake of simplicity in deriving its Fourier series, let us assume that the fundamental frequency of the estimate is f_o and not a submultiple of it. This means that $\hat{x}(k)$ periodically takes on P discrete values, denoted as $\hat{x}_j$, every T_o seconds. In this analysis, the estimate takes the form of a staircase-like waveform and each discrete value, $\hat{x}_j$, lasts for T_o/P seconds. Using the continuous notation, $\hat{x}(t)$, the Fourier series of the DM estimate can be expressed as

$$\hat{x}(t) = C_o + \sum_{n=1}^{\infty} C_n \cos(2\pi n f_o t + \phi_n) \ , \tag{2.8-3a}$$

where

$$C_o = (1/T_o)\int_0^{T_o} \hat{x}(t)\,dt \ , \tag{2.8-3b}$$

$$C_n = \sqrt{A_n^2 + B_n^2} , \tag{2.8-3c}$$

$$\phi_n = -\arctan(B_n/A_n) , \tag{2.8-3d}$$

and

$$A_n = (2/T_o)\int_0^{T_o} \hat{x}(t)\cos(2\pi n f_o t)\,dt , \tag{2.8-3e}$$

$$B_n = (2/T_o)\int_0^{T_o} \hat{x}(t)\sin(2\pi n f_o t)\,dt . \tag{2.8-3f}$$

Substituting the discrete values of $\hat{x}(t)$, we can rewrite Eqs. (2.8-3e) and (2.8-3f) as

$$A_n = (2/T_o)\sum_{j=1}^{P} \hat{x}_j \int_{(j-1)T_o/P}^{jT_o/P} \cos(2\pi n f_o t)\,dt \tag{2.8-4a}$$

$$B_n = (2/T_o)\sum_{j=1}^{P} \hat{x}_j \int_{(j-1)T_o/P}^{jT_o/P} \sin(2\pi n f_o t)\,dt . \tag{2.8-4b}$$

By invoking some trigonometric identities, we can further reduce A_n and B_n to a form which is readily adaptable for computer simulation, that is,

$$A_n = \frac{2\sin(n\pi/P)}{n\pi} \sum_{j=1}^{P} \hat{x}_j \cos[n\pi(2j - 1)/P] \tag{2.8-5a}$$

$$B_n = \frac{2\sin(n\pi/P)}{n\pi} \sum_{j=1}^{P} \hat{x}_j \sin[n\pi(2j - 1)/P] . \tag{2.8-5b}$$

Now we have available the strength of the Fourier components of the DM estimate, C_n, and thus the essential information to evaluate the DM performance. Since we are ultimately concerned with the performance of the DM processors, it remains to extend this development to the DM sum and the DM product.

Similar to the case when we considered the response of the DM adder and multiplier to constant inputs, we need only refer to the basic initial design equations, i.e., Eq. (2.2-1) and Eq. (2.4-1). These tell us that the direct sum and product are formulated as the sum and product, respectively, of the individual signal estimates. Since we shall be adding or multiplying two periodic sinusoidal steady state patterns, $a_D(k)$ and $p_D(k)$ must also be periodic signals. Therefore, the Fourier series representation theory is immediately applicable to the DM direct sum and product.

2.8.2 Output SNR

To determine the desired figure of merit, we must go from the Fourier series of the DM estimate to the output SNR. Since we are concerned with an audio mode DM, it would seem reasonable to choose a low pass filter (LPF) applicable to voice signals to bridge this gap. A LPF commonly used in experimental work is a fourth-order Butterworth type whose magnitude-squared transfer function is given as,

$$|H_B(s)|^2 = 1/[1 + (s/\omega_c)^8] \ , \qquad (2.8\text{-}6)$$

where

$\omega_c \equiv$ the radian cutoff frequency.

The frequency characteristics of this LPF are:

$$|H_B(f)|^2 = 1/[1 + (f/f_c)^8] \ , \qquad (2.8\text{-}7)$$

where

$$f_c = \omega_c/2\pi \ . \qquad (2.8\text{-}8)$$

To realistically represent a voice signal, we shall choose $f_o = 800$ Hz and $f_c = 4f_o = 3200$ Hz. From Eq. (2.8-7) we can obtain the attenuation factor, α_n, that we must scale the Fourier components of $\hat{x}(t)$ by to simulate low pass filtering. Using $f_c = 4f_o$, we find that

$$\alpha_n = [1 + (n/4)^8]^{-\frac{1}{2}} \ . \qquad (2.8\text{-}9)$$

Since all harmonics are orthogonal, we shall be concerned only with the attenuation produced by the LPF and not consider the phase shift which arises.

After final low pass filtering, the output signal power becomes

$$S_o = \tfrac{1}{2}(\alpha_1 C_1)^2 \ . \qquad (2.8\text{-}10)$$

The output noise power comes from all the filtered frequency harmonics other than the fundamental. After the LPF, the output noise power is expressable as

$$N_o = \frac{1}{2} \sum_{n=2}^{\infty} (\alpha_n C_n)^2 \,. \qquad (2.8\text{-}11)$$

Thus, the output signal-to-noise ratio (SNR_o) is given as

$$SNR_o = \frac{S_o}{N_o} = \frac{(\alpha_1 C_1)^2}{\sum_{n=2}^{\infty} (\alpha_n C_n)^2} \,. \qquad (2.8\text{-}12)$$

2.8.3 Performance Curves

From our system simulations we verified that the DM estimate, the direct DM sum and the direct DM product all produced periodic responses to sinusoidal inputs when the DM bit rate was an integral multiple of the frequency of the sinusoid. By expanding the simulation programs we were able to incorporate the calculation of the Fourier components of the various outputs. All the simulations, including calculation of output SNR, were carried out on the PDP 8/L computer using the FOCAL system. FOCAL, an abbreviation for Formulating On-line Calculations in Algebraic Language, is a conversational programming language which is similar to, but not as powerful as FORTRAN or BASIC.

In determining SNR_o, we did not use an infinite number of harmonics to calculate the noise power as required by Eq. (2.8-11). Instead, we truncated after the ninth harmonic since $(\alpha_{10}C_{10})^2$ was negligible in comparison with the noise due to the second through ninth harmonics. We have found that, for the same input signal power and for the same

ratio, f_s/f_o, the periodic sinusoidal steady state pattern that the DM estimate assumes, and consequently the SNR_o, are very dependent upon the starting point of the input sinusoid. Since the input is zero before the sinusoid starts and the ADM estimate tracks the zero input with a periodic pattern of duration $4T_s$, the sinusoid can start at any point within this interval with equal probability. The SNR_o has become a random variable dependent on the starting point of the input sine wave. In Fig. 2.8-1, we show the ADM response to a constant input and a number of possible starting points of the input sinusoid. To obtain a truly representative value of SNR_o we employed 40 different starting points and calculated the mean and the standard deviation about the mean, $\sigma(SNR_o)$.

We have generated several families of performance curves. In Fig. 2.8-2, we show the SNR in dB for the direct sum, SNR_a, versus relative input signal power over a range of 54 dB for ratios of $f_s/f_o = 60$, 40 and 20. This ratio is equivalent to the $f_s/R_a = 7.5$, 5 and 2.5, where the variable R_a is the PCM Nyquist rate for the sum. Thus, f_s/R_a is the number of bits in an equivalent PCM system. The input signal power was varied by changing the amplitude of the input sinusoid. A relative input signal power of -6 dB corresponds to an amplitude of 5S, where S is the minimum step size, while 42 dB represents an amplitude of 1280S.

To obtain the performance curves for the direct product, we let both inputs equal the same sinusoid. Therefore, the

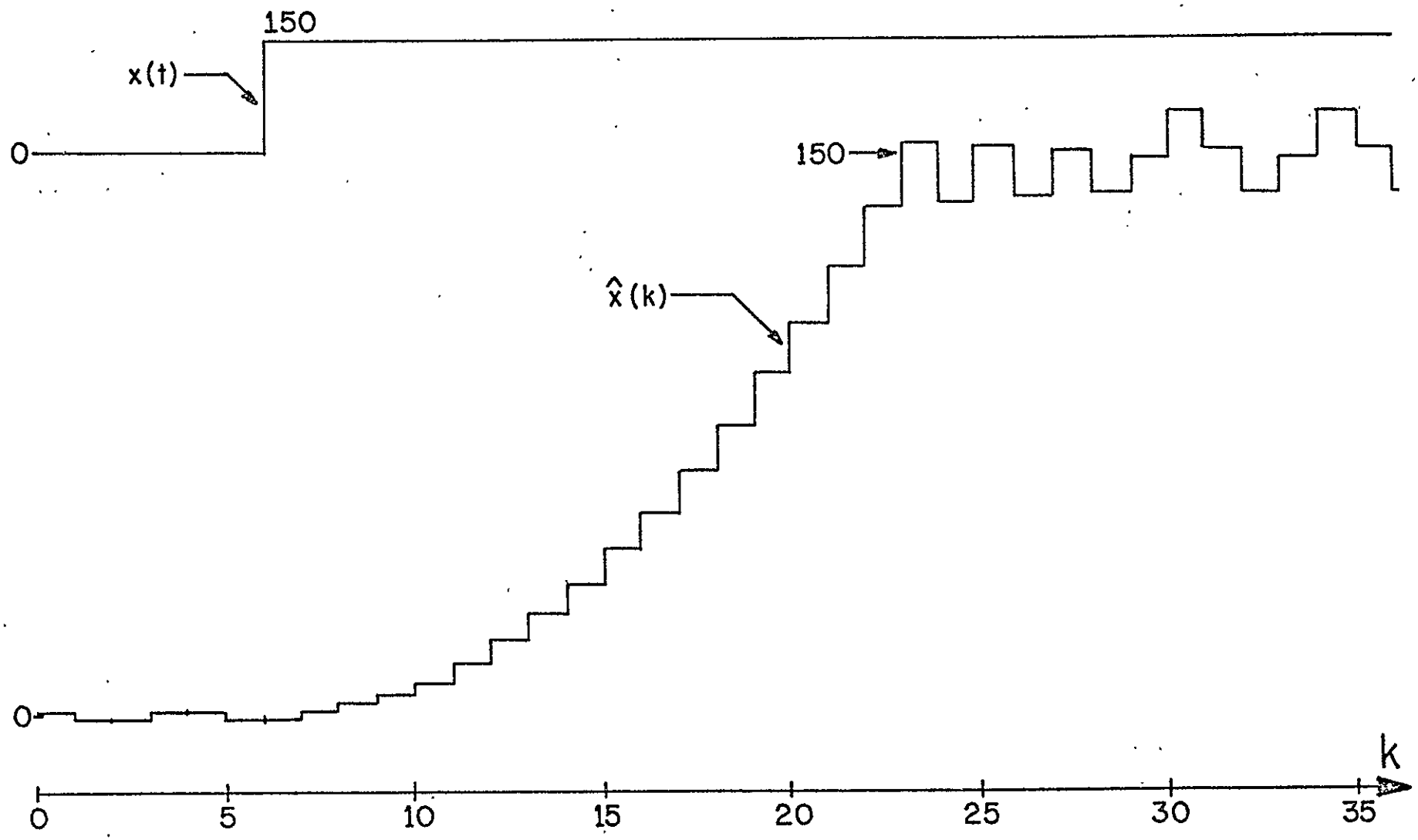


Fig. 2.7-8. Song Audio Mode DM Response to a Step

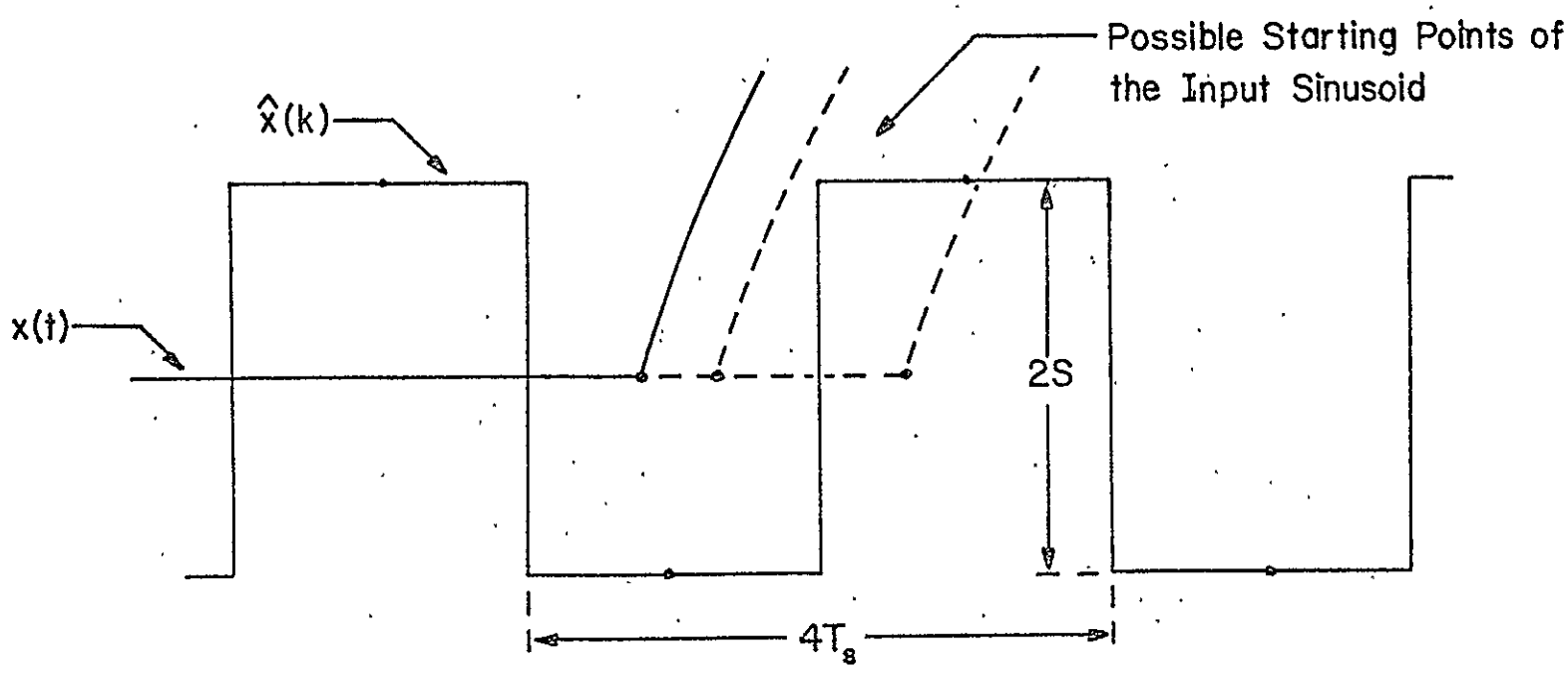


Fig. 2.8-1. Song Audio Mode ADM Response to a Constant Input

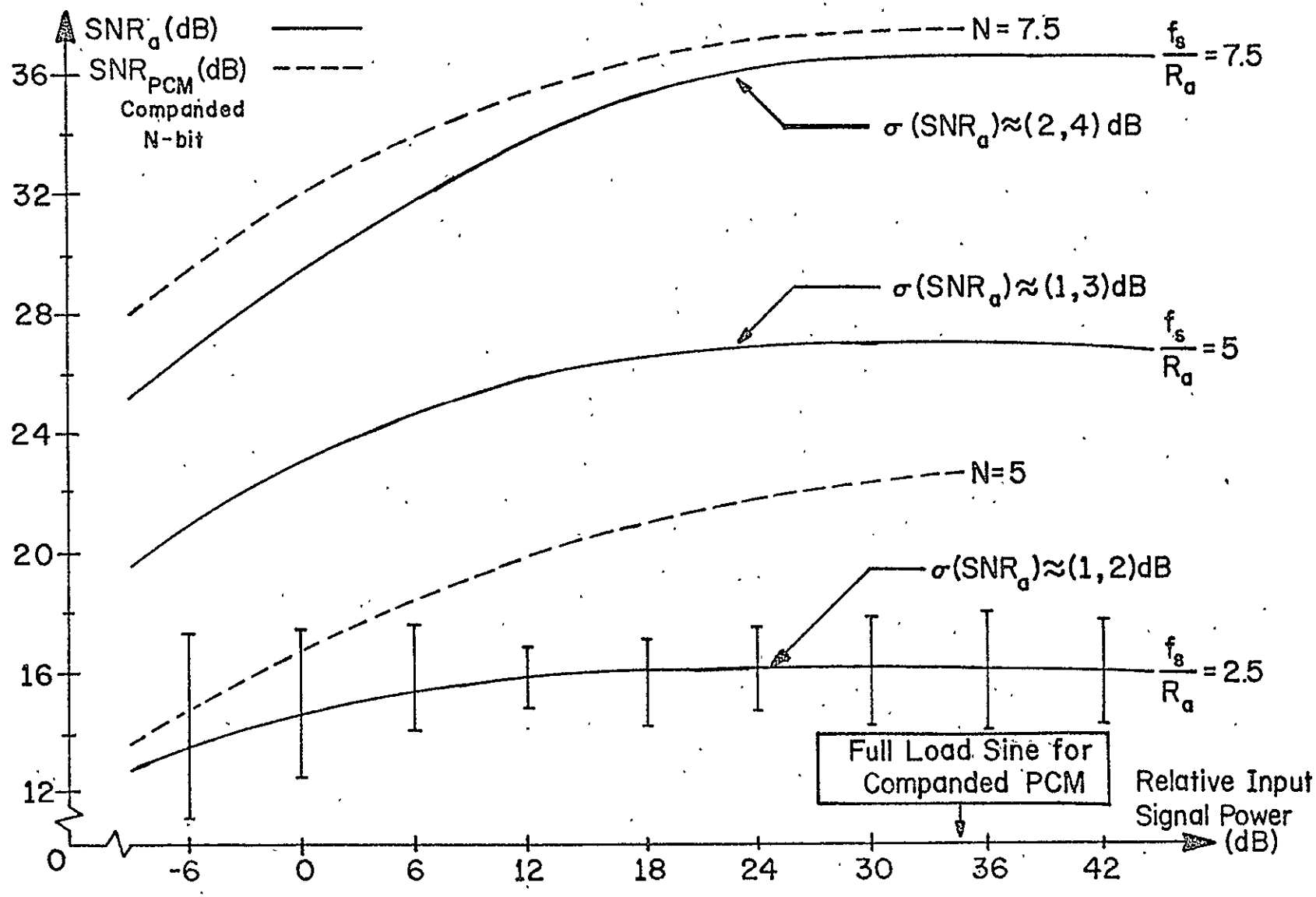


Fig. 2.8-2. SNR for DM Direct Sum

product becomes the square of the input signal. In this case, the output signal power comes from the product frequency, $f_p = 2f_o$, and the noise from all other harmonics. In Fig. 2.8-3, we give the SNR for the direct product, SNR_p, as a function of relative input signal power over the same range of 54 dB with the parameter f_s/f_p taking on values 30, 20 and 10. This parameter is the same as the ratio $f_s/2R_p = 3.75$, 2.5 and 1.25, where R_p is the PCM Nyquist rate for the product. We shall see that $f_s/2R_p$ is the number of input bits in an equivalent PCM multiplier. Since the output signal frequency is twice the input signal frequency, we naturally expect SNR_p to be less than the SNR from the DM sum because the DM is frequency sensitive.

Recall that the SNR is actually a random variable, as explained previously. Consequently, all the plots shown above were drawn as smooth curves through windows of one standard deviation about the mean value of the SNR. On the SNR_a family we show the set of windows for one curve. In lieu of the other standard deviation windows, we denote the range of the standard deviation for each curve in both the sum family and the product family. The significance of these performance curves will be detailed in the next section when we compare the DM systems to PCM processors.

2.9 Comparison with PCM Systems

We shall compare the DM direct processors with their dual PCM systems based on the SNR achieved with equivalent

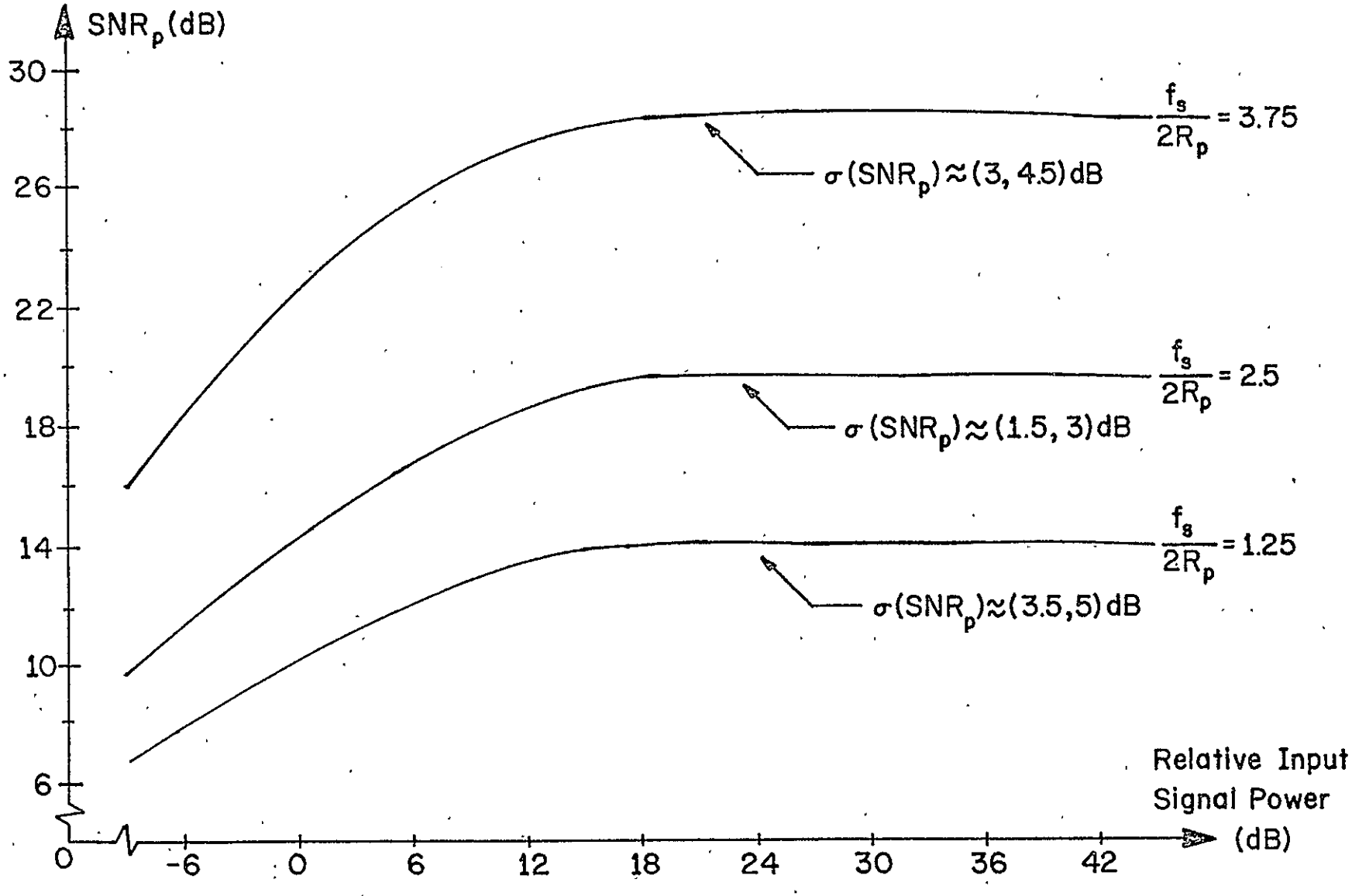


Fig. 2.8-3. SNR for DM Direct Product

channel bit rates. To achieve a wide dynamic range, <u>companded</u> PCM is generally used for encoding speech signals. With an input sinusoid, the SNR for companded PCM is given as

$$\mathrm{SNR(PCM)} = 6N + 1.8 + I(\sigma_i^2) \quad \mathrm{dB}\ , \qquad (2.9\text{-}1a)$$

where

N = the number of bits used to PCM encode the input signal

and

$I(\sigma_i^2)$ = the companding improvement factor, a function of the input signal power.

For logarithmic companding commonly used by the telephone company, a family of improvement curves is given which displays $I(\sigma_i^2)$ as a function of signal power below full load sinusoid for several values of the compression factor, μ [17]. A full load sinusoid has the maximum amplitude that can be encoded with N bits.

If we choose a median value for the compression factor ($\mu = 100$), then, for a full load sinusoid, $I \simeq -9.5$ dB. Thus, the maximum SNR for companded PCM with $\mu = 100$ is

$$\mathrm{SNR_m(PCM)} = 6N - 7.7 \quad \mathrm{dB}\ . \qquad (2.9\text{-}1b)$$

Since this compression factor yields a SNR curve which drops off only 3 dB from the maximum value over a dynamic range of 30 dB, Eq. (2.9-1b) will be our basis of comparison for companded PCM systems.

For the case of digital addition, for both PCM and DM, it does not matter if we encode, process and transmit or encode, transmit and process. There is no change in channel bit rate. This point will be critical when comparing multiplication. To best explain the comparison with the PCM addition system, we shall refer to the chart given in Table 2.9-1. Consider the bandwidth of each input signal to be B Hz. The input signals are encoded into N-bit PCM and then processed. Since we are adding the two signals, the output bandwidth is also B Hz. The PCM sampling rate for the sum is

$$R_a = 2B \ , \tag{2.9-2}$$

and the number of bits it will have, assuming no overflow, is

$$N_a = N \ . \tag{2.9-3}$$

Consequently, the PCM channel bit rate becomes

$$R_a N_a = 2BN \ . \tag{2.9-4}$$

Since we use an input tone of frequency f_o in determining performance curves and set the output LPF to cutoff at $4f_o$, we must equate the cutoff frequency to the output bandwidth, i.e.,

$$f_c = 4f_o = B \ . \tag{2.9-5}$$

To compare the systems, we specify equivalent channel

PERFORMANCE PARAMETERS FOR COMPARING DM ENCODED $a_D(k)$ & $p_D(k)$ TO N-BIT PCM		
	SUM	PRODUCT
Input Signal Bandwidth	B	B
Output Signal Bandwidth	B	2B
PCM Sampling Rate	$R_a = 2B$	$R_p = 4B$
PCM Bits in Processor	$N_a = N$	$N_p = 2N$
PCM Channel Rate	2BN	8BN
Input Tone Frequency	f_o	f_o
Output LPF Cutoff Frequency	$f_c = 4f_o = B$	$f_c = 4f_o = 2B$
DM Sampling Rate	$f_s = 2BN = NR_a$	$f_s = 8BN = 2NR_p$
DM SNR Index	$f_s/R_a = N$	$f_s/2R_p = N$
DM Performance Curves Sampling Rates	$f_s = 20f_o = 2.5R_a$	$f_s = 20f_o = 1.25(2R_p)$
	$f_s = 40f_o = 5R_a$	$f_s = 40f_o = 2.5(2R_p)$
	$f_s = 60f_o = 7.5R_a$	$f_s = 60f_o = 3.75(2R_p)$

TABLE 2.9-1. Comparison of DM and PCM Arithmetic Processors

bit rate. For the DM adder, the bit rate is f_s. Therefore,

$$f_s = 2BN = R_a N \ . \qquad (2.9\text{-}6)$$

The DM SNR index used in Fig. 2.8-2 is clearly the number of PCM bits used to encode the input signals, i.e.,

$$f_s/R_a = N \ . \qquad (2.9\text{-}7)$$

On the performance graph for the DM direct sum, Fig. 2.8-2, we have plotted the SNR for N-bit companded PCM, when N = 7.5 and 5. For these curves, we set the full load amplitude equal to 512S, representing a DM system with 10 bits of internal arithmetic. In these two cases, the PCM and DM systems yield comparable performance. When N = 2.5, SNR_m(PCM) = 7.3 dB in comparison with approximately 16 dB for the ADM adder. At low transmission rates, the DM system clearly has the advantage.

When considering digital multiplication, it is very important to process first and then to transmit. If we reverse this order, the DM bit rate will increase by a factor of two and the resulting performance will appear that much downgraded. As in the case of addition, we shall refer to Table 2.9-1 extensively in this comparison. Again, the input signals are bandlimited to B Hz and encoded into N-bit PCM signals. The PCM multiplication causes the output signal to have a bandwidth of 2B and consequently, the sampling rate for the product must be

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

$$R_p = 4B \ . \tag{2.9-8}$$

More important, the number of bits needed for the product is

$$N_p = 2N \ . \tag{2.9-9}$$

Thus, the channel bit rate is

$$R_p N_p = 8BN \ . \tag{2.9-10}$$

Again, the frequency of the input tone is f_o and the LPF cutoff frequency is set to $4f_o$. Equating the output bandwidth to the cutoff frequency, we obtain

$$f_c = 4f_o = 2B \ . \tag{2.9-11}$$

Now we can compare the systems on the basis of equal channel bit rate. Since we are transmitting the DM encoded direct product the bit rate is still f_s. Therefore,

$$f_s = 8BN = 2R_p N \ . \tag{2.9-12}$$

Now it is obvious that the parameter employed in the DM product SNR graph is the number of bits used to PCM encode the input signals, that is,

$$f_s/2R_p = N \ . \tag{2.9-13}$$

Comparing the performance curves shown in Fig. 2.8-3 with Eq. (2.9-1b), we find that the DM SNR is consistently higher. We must remember that because we have DM encoded $p_D(k)$ in this comparison system, we should really look at

the SNR curves of $\hat{p}_D(k)$. But these curves are exactly the same as Fig. 2.8-3 except 1-2 dB lower. This is the same effect as experienced when placing DM links in tandem [18].

We have shown the feasibility of adding and multiplying DM encoded signals. The systems presented use only standard digital devices in their realizations and the performance curves do not show any surprising results. Comparing the DM processors to PCM systems for the same bit rate, we have shown that their performance is either comparable or better. Although the order of operation is unimportant when adding, it is vital to first process and then transmit when multiplying.

As a concluding remark, we shall consider some practical aspects of the DM processors. The DM adder can be applied when we need to mix voice channels, as in a stereo system. The DM multiplier can serve as the basis for constructing a correlator. Although we can realize only discrete time delays, they take the very simple form of one-bit shift registers, since we must merely delay $e_x(k)$.

A final consideration is the measurement of output SNR. With a DM system that has to be physically constructed, the SNR will not be a random variable dependent on the starting point of the input sinusoid. Consequently, we will not measure different values of SNR each time the device or the input is turned on. In a real system, we can never guarantee that f_s will be an exact integral multiple of f_o. In fact, the DM clock and an input sinusoid are actually non-

coherent signals. This noncoherence has been modeled as different starting points. The measured output SNR is therefore analogous to the mean SNR with an infinite number of different initial conditions.

CHAPTER 3

CONVERSION FROM ADM ENCODED SIGNALS TO PCM ENCODED SIGNALS USING DIGITAL FILTER TECHNIQUES

Due to high quality performance and ease in implementation, many modern communications systems are employing digital encoding techniques. Among the existing digital encoding techniques, ADM and PCM are widely utilized in commercial communications. To facilitate digital processing of signals encoded in ADM and PCM formats, there is a need for translation units between the two systems. In this chapter, we consider conversion from ADM to PCM format.

A general technique is presented for converting ADM encoded signals to PCM format without first demodulating the ADM bit stream and returning to the analog domain. The translation unit that is derived employs only standard digital hardware and is applicable to a large class of ADM encoders. SNR curves are given for PCM converted, sinusoidal signals obtained from the three different systems which are presented in this chapter. Thus, we can present a relevant evaluation of the performance of these ADM to PCM converters.

3.1 Basic ADM-PCM Conversion Philosophy

Consider an analog signal, $x(t)$, assumed bandlimited to f_m, which has been ADM encoded at a rate f_s. The general form of the ADM was given in Fig. 2.1-1 and mathematically

described by Eqs. (2.1-1) through (2.1-4). The only constraint imposed upon this system is that the ADM bit rate is an integral multiple of the Nyquist rate, that is,

$$f_s = Rf_N, \tag{3.1-1}$$

where

$$f_N = 2f_m \tag{3.1-2}$$

and

$$R = \text{a positive integer greater than 1.}$$

In our conversion system, the value of R is set by f_s and we cannot vary this in our design. Since we are converting between two digital encoding systems, we restrict our design to an all-digital technique which can be implemented with standard digital hardware.

The conversion from ADM to PCM encoded signals entails changing from a high "information" rate (ADM) to a lower one (PCM). Formally stated, an ADM to PCM converter operates on the sequence $\{e_x(k)\}$ and produces x(t) in PCM format. Examining the ADM in Fig. 2.1-1, we see that one of its basic equations is

$$x(k) = \hat{x}(k) + \xi x(k) \ , \tag{3.1-3}$$

where

$x(k)$ = the input signal,
$\hat{x}(k)$ = the ADM estimate

and

$\xi_x(k)$ = the error signal.

Since the ADM is operating at a rate f_s, the above notation

means that

$$x(k) = x(t = kT_s) \quad , \tag{3.1-4}$$

where

$$T_s = 1/f_s \quad . \tag{3.1-5}$$

Usually, $x(k)$ is given and we find $\hat{x}(k)$. Now the problem is reversed: $\hat{x}(k)$ is given, via the ADM bits, $\{e_x(k)\}$, and we want to find $x(t)$. To do this, we must estimate $\xi_x(k)$. Then, to achieve PCM format, we must sample $\hat{x}(t) + \xi_x(k)$ at a rate $f_N = f_s/R$. Consequently, the ADM to PCM converter should perform the function:

$$x(Rk) = \hat{x}(Rk) + \xi_x(Rk) \quad , \tag{3.1-6}$$

where $\hat{x}(k)$ and $\xi_x(k)$ are formed from the ADM bits and then they are sampled at the PCM rate, f_N. We observe that

$$x(Rk) = x(t = kT_N) = x(t = RkT_s) \tag{3.1-7}$$

since

$$T_N = 1/f_N = RT_s \quad . \tag{3.1-8}$$

To obtain $\xi_x(k)$ from $e_x(k)$, we must realize the inverse of a hard limiter. But, there is no physically realizable one-to-one inverse of a hard limiter. The information lost going from $\xi_x(k)$ to $e_x(k)$ cannot be recovered. However, the basic idea of improving the ADM estimate is still a valid concept. This is exactly the problem that we are faced with, that is, how to improve $\hat{x}(k)$ before we decrease the "information" rate from f_s to f_N by sampling at the Nyquist

rate.

The PCM values obtained in this way are in a linear PCM format. However, most existing PCM systems use a companded code. To achieve compatibility, we can use a simple ROM as a digital code converter since there is a one-to-one mapping between linear and companded PCM.

3.2 ADM Decoder Technique

The ADM to PCM converters that will be discussed are applicable to any ADM that can be described with Eqs. (2.1-1) through (2.1-4). However, when simulating and obtaining performance curves, we must use a particular ADM mode. Since we are confining our applications to speech signals, all references to the ADM step size will apply to the Song audio mode algorithm given in Eq. (2.1-6).

The simplest and most obvious method of converting an ADM encoded signal to a PCM encoded signal is to decode $\{e_x(k)\}$ by generating and accumulating the step sizes, $S_x(k)$, to form the ADM estimate, $\hat{x}(k)$. By the very nature of the DM digital feedback circuit, $\hat{x}(k)$ is a parallel digital word. Since the DM operates at a rate much higher than the Nyquist rate, the estimate must be resampled, via a bank of AND gates, at the Nyquist rate, to yield PCM words, $\hat{x}(Rk)$. This device, called the ADM estimate converter, is shown in Fig. 3.1-1. In this converter, we have set $\xi_{\hat{x}}(Rk) = 0$ in Eq. (3.1-6) and, therefore, we have not improved $\hat{x}(Rk)$ at all.

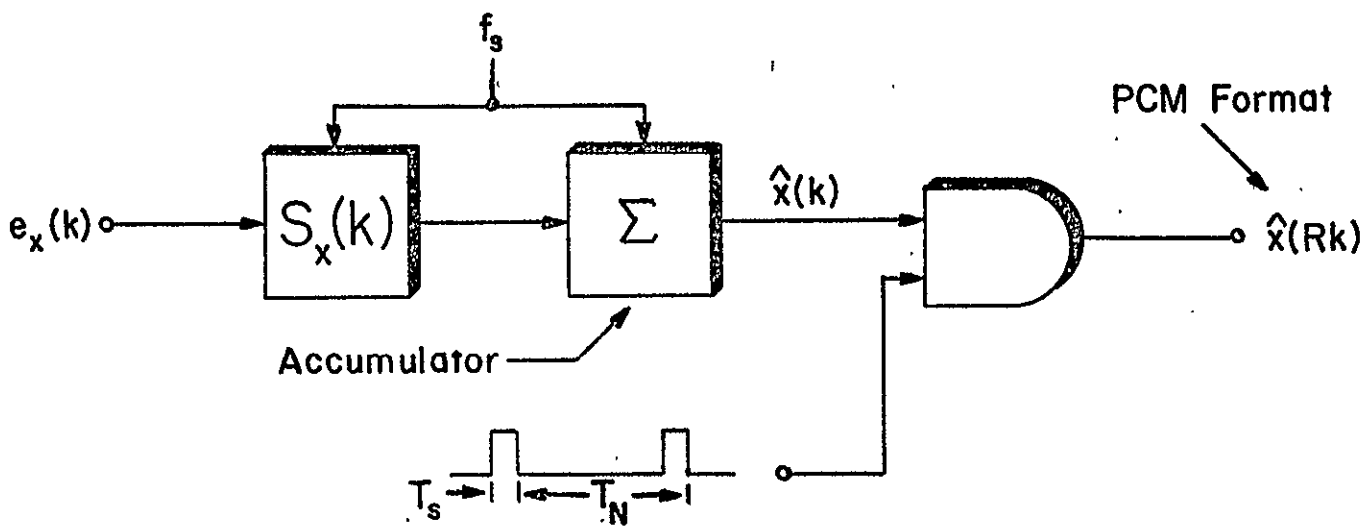


Fig. 3.1-1. ADM Estimate Converter

Heuristically, we can argue that this technique will not produce a high quality PCM signal regardless of the number of bits used in the accumulator to form $\hat{x}(k)$. This is due to the high probability of choosing a "poor" value of $\hat{x}(k)$, i.e., a sample yielding a large error between it and the true value, $x(k)$. A "poor" ADM estimate is often produced when the step size has grown too large causing the estimate to overshoot the input signal. If the input continues to increase, the estimate must reverse direction for one ADM period, due to the overshoot, before it again continues to increase. The one period, where the estimate has reversed direction, generally yields an extremely "poor" value of $\hat{x}(k)$.

Consequently, although we can obtain a high quality representation of the original signal by analog low pass filtering the entire ADM estimate, the same is not true if we filter samples of the estimate taken at the Nyquist rate. When $\hat{x}(k)$ is passed through a LPF, the "poor" estimate values, occurring for only T_s seconds, are easily averaged out because the LPF cutoff frequency is much less than f_s. However, since the PCM samples occur at the Nyquist rate, a "poor" value will give rise to a considerable error, even after final low pass filtering. This has been verified through computer simulation and will be presented later. The analog LPF is used to return to the analog domain, where performance is evaluated and for no other purpose.

The conclusion of this heuristic argument can be further

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

strengthened with a frequency domain analysis of the ADM estimate converter. The ADM encoding process, even when operating on an initially bandlimited signal, always generates out-of-band frequency components in the estimate. If we let M(f) represent the spectrum of $\hat{x}(k)$, then M(f) will not be bandlimited to f_m. In this converter, the basic operation that must be performed to achieve a PCM format is sampling at the Nyquist rate, f_N. The sampling operation causes a shifting of M(f) along the frequency axis. The spectrum of $\hat{x}(k)$ sampled at the Nyquist rate is given as,

$$M_N(f) = M(f) + \sum_{i=1}^{\infty} [M(f + if_N) + M(f - if_N)] \quad , \quad (3.2\text{-}3)$$

where the above additions are performed vectorially since M(f) is a complex quantity. The result of sampling a non-bandlimited signal is, of course, aliasing. It is precisely this fact that causes distortion in the resultant PCM signal and, therefore, a low quality representation of the original signal which, as we shall see later, manifests itself as a low SNR on our performance curves.

To eliminate the "poor" values of $\hat{x}(k)$ and still maintain a completely discrete system, we can insert a digital LPF after the DM estimate, just before the gating device operating at the Nyquist rate. This digital filter may be viewed as a device which filters in the frequency domain, produces a statistical estimate or performs a digital interpolation. In all cases, it will decrease the out-of-band noise and make all the estimate values accurate before we

resample. This concept will be further pursued in the next section.

3.3 Non-Recursive Digital LPF Technique

The objective of the ADM to PCM converter is to eliminate the "poor" values of $\hat{x}(k)$ before resampling. Since these "poor" values are surrounded by many "good" values of $\hat{x}(k)$, some averaging or filtering of $\hat{x}(k)$ before Nyquist sampling should improve the PCM signal. The method presented to achieve this objective is an extension of the concept originally proposed by D. Goodman [19] to perform analog to PCM conversion using a DM as an intermediate step. We now apply this technique to a general class of ADMs rather than to merely a linear DM which was done in this reference. Goodman used a minimum mean-square error criterion to determine the coefficients in the non-recursive digital filter. To complete the design, it was necessary to assume input signal statistics. In our system, we employ a method to determine the filter coefficients which is completely independent of input signal statistics. Our design is therefore robust.

To improve the PCM converted signal over that obtained from the system shown in Fig. 3.1-1, we insert a low pass filter after the accumulator to eliminate the spurious frequency components of the signal estimate. The ADM to PCM converter now has the step size, $S_x(k)$, acting as the input of two cascaded, linear filters. The latter of these linear

filters is sampled at the Nyquist rate to produce improved PCM samples, $\tilde{x}(Rk)$. The block diagram of this system is given in Fig. 3.3-1. The accumulator is represented as an ideal integrator, whose impulse response is

$$a(t) = 1, \quad t \geq 0 \ , \qquad (3.3\text{-}1)$$
$$= 0, \quad t < 0 \ ,$$

and the LPF is designated by its impulse response, $h(t)$.

Since both $a(t)$ and $h(t)$ represent linear systems, they can be combined into one linear filter, via convolution, i.e.,

$$g(t) = a(t) * h(t) = \int_{-\infty}^{t} a(t - \lambda)h(\lambda)d\lambda \ , \qquad (3.3\text{-}2)$$

where the upper limit is due to the causality of the accumulator, that is, $a(t - \lambda) = 0$ for $\lambda > t$. For the limits of integration in Eq. (3.2-2), $a(t - \lambda) = 1$, and therefore

$$g(t) = \int_{-\infty}^{t} h(\lambda)d\lambda \ , \qquad (3.3\text{-}3)$$

which is merely the unit step response of the LPF. The improved ADM estimate, $\tilde{x}(k)$, can be formulated from the following discrete convolution,

$$\tilde{x}(k) = \sum_{j=-\infty}^{\infty} S_x(k - j)g(j) \ , \qquad (3.3\text{-}4)$$

where

$$g(j) = g(jT_s)$$

and

$$T_s = \text{the ADM sampling period.}$$

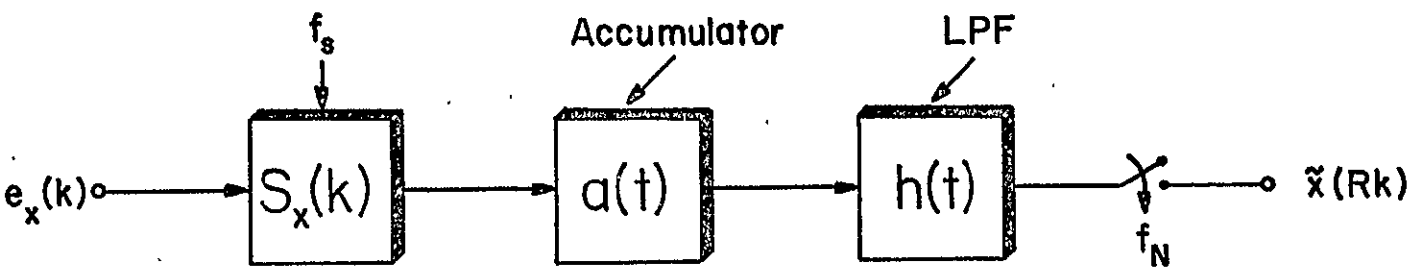


Fig. 3.3-1. Improved ADM Estimate Converter

Since $g(t)$ represents the unit step response of a LPF, we know that

$$\lim_{j\to\infty} g(j) = 1 \ , \tag{3.3-5a}$$

and there exists a value of j for which $g(j)$ is arbitrarily close to 1. If we designate this value of j as Q, then we have

$$g(j) \simeq 1, \text{ for } j \geq Q \ . \tag{3.3-5b}$$

The improved ADM estimate can therefore be approximated very closely by

$$\tilde{x}(k) = \sum_{j=-\infty}^{Q-1} S_x(k - j)g(j) + \sum_{j=Q}^{\infty} S_x(k - j) \ . \tag{3.3-6}$$

If we change the index in the second sum, letting $i = k - j$, then

$$\sum_{j=Q}^{\infty} S_x(k - j) = \sum_{i=-\infty}^{k-Q} S_x(i) = \hat{x}(k - Q) \ . \tag{3.3-7}$$

Specifying the LPF as being causal so that $h(t) = 0$ for $t < 0$ and, consequently, $g(t) = 0$ for $t < 0$, the final discrete form of $\tilde{x}(k)$ is given as

$$\tilde{x}(k) = \sum_{j=0}^{Q-1} g(j)S_x(k - j) + \hat{x}(k - Q) \ . \tag{3.3-8}$$

From Eq. (3.3-8), we observe that the original ADM estimate, $\hat{x}(k - Q)$, is modified by the addition of a weighted sum of past step sizes. The PCM samples now take the exact form suggested by Eq. (3.1-6), except for a time delay in

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

the ADM estimate. This entire system is depicted in Fig. 3.3-2, where we show an ADM to PCM converter with a non-recursive digital filter. The advantages of this configuration are that the filter is absolutely stable, the coefficients of the filter can deviate from the exact value without drastically affecting the filter frequency characteristics, and, as pointed out in a somewhat similar practical realization of this concept [20], the hardware capacity needed for filtering $S_x(k)$ is much less than that needed if we operated on the estimate, $\hat{x}(k)$, with a non-recursive filter.

In the realization of the ADM to PCM converter using a non-recursive filter, there are several practical considerations that should be pointed out. Since the filter coefficients do not require extreme precision, we may be able to realize the products $g(j)S_x(k - j)$ by employing hard-wired scalers as discussed in Sec. 2.2. Thus, we eliminate the need for digital multipliers and reduce the hardware complexity of the system.

The hardware structure for this converter can be completely modified by recursively realizing the product $g(j)S_x(k - j)$ from the ADM bit stream. Let us define this term as

$$v_j(k) \equiv g(j)S_x(k - j) \ . \tag{3.3-9}$$

Using the Song audio mode algorithm for the step size, Eq. (2.1-6), the product becomes

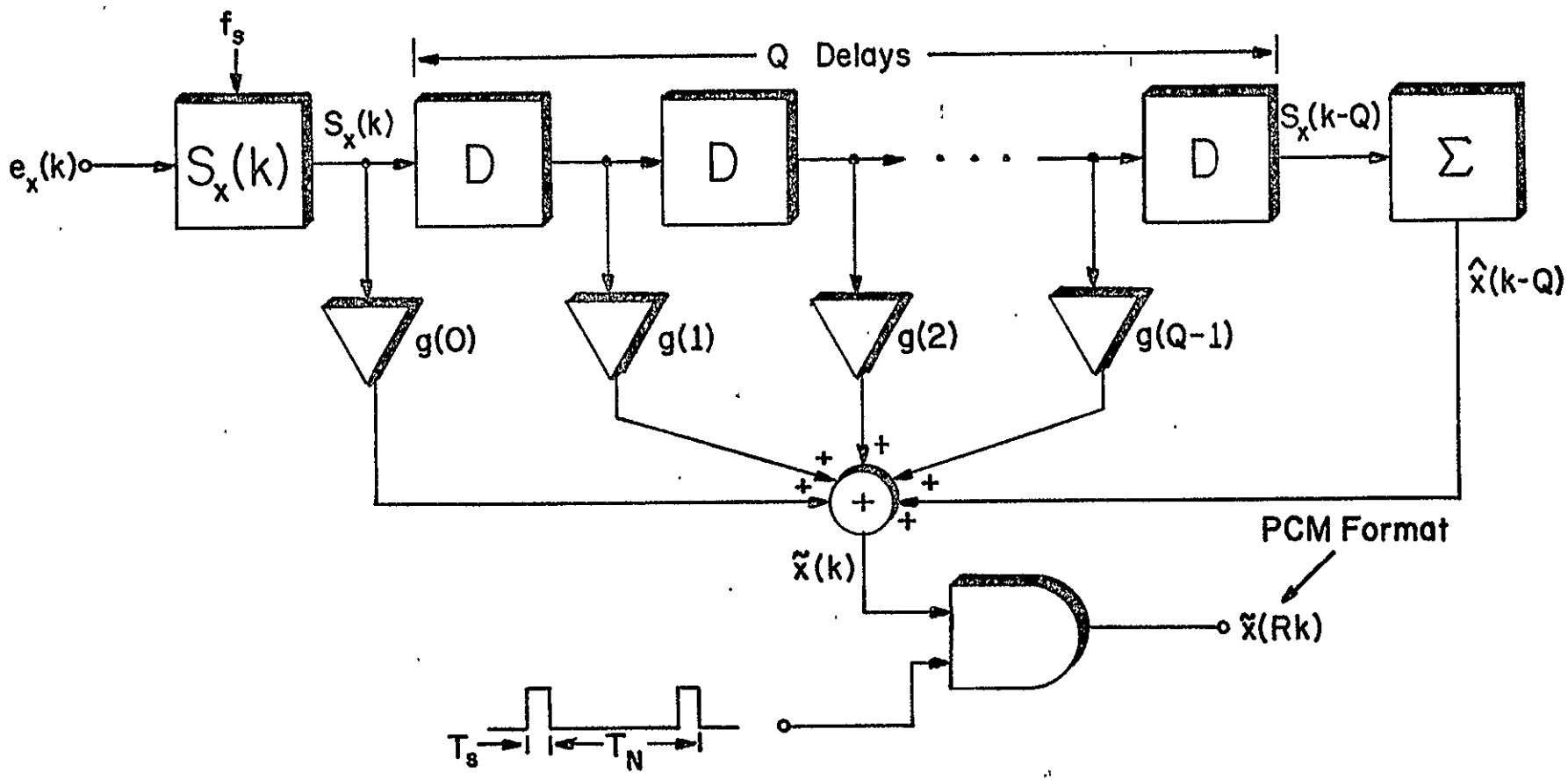


Fig. 3.3-2. ADM to PCM Converter with Non-Recursive Filter

$$v_j(k) = g(j)|S_x(k - j - 1)|e_x(k - j - 1) + g(j)Se_x(k - j - 2). \tag{3.3-10}$$

If we employ the step size relationship given in Eq. (2.4-4), we obtain

$$v_j(k) = g(j)S_x(k - j - 1)e_x(k - j - 1)e_x(k - j - 2) + g(j)Se_x(k - j - 2). \tag{3.3-11}$$

Expressed recursively, the product now becomes

$$v_j(k) = v_j(k - 1)e_x(k - j - 1)e_x(k - j - 2) + g(j)Se_x(k - j - 2). \tag{3.3-12}$$

The block diagram realization of $v_j(k)$ is shown in Fig. (3.3-3).

Returning to Eq. (3.3-8), the improved estimate can be formulated as

$$\tilde{x}(k) = \sum_{j=0}^{Q-1} v_j(k) + \hat{x}(k - Q) . \tag{3.3-13}$$

The block diagram corresponding to this equation for the ADM to PCM converter with non-recursive filter is given in Fig. (3.3-4). Although there is no saving in hardware with this modified realization, the accuracy of the product is improved. This can be seen from Fig. 3.3-3. Even though we still must scale by g(j), here we are scaling one minimum step size rather than $S_x(k - j)$. Even if the numerical scale factor is a gross approximation to the true value, the most

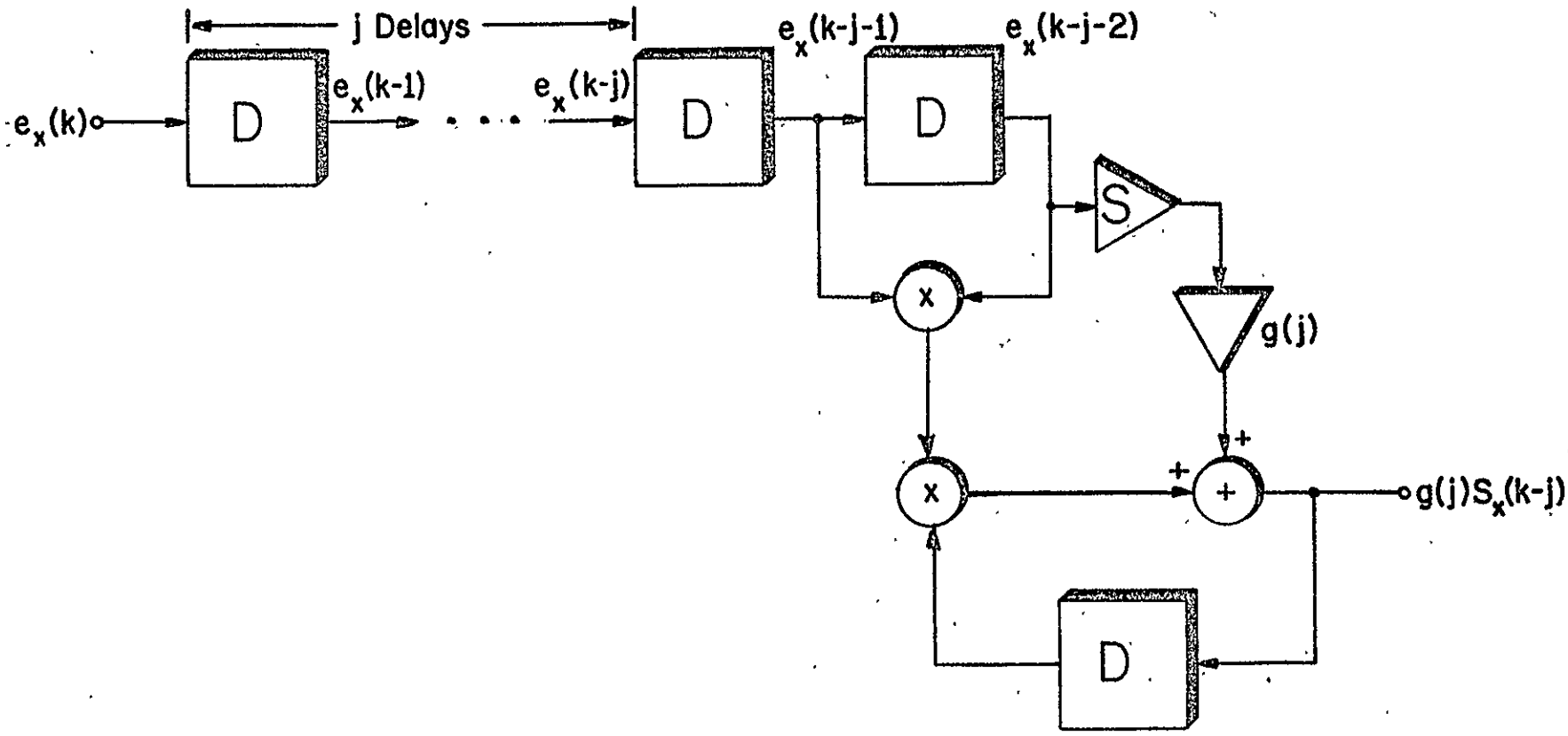


Fig. 3.3-3. Realization of the Product $g(j)S_x(k-j)$

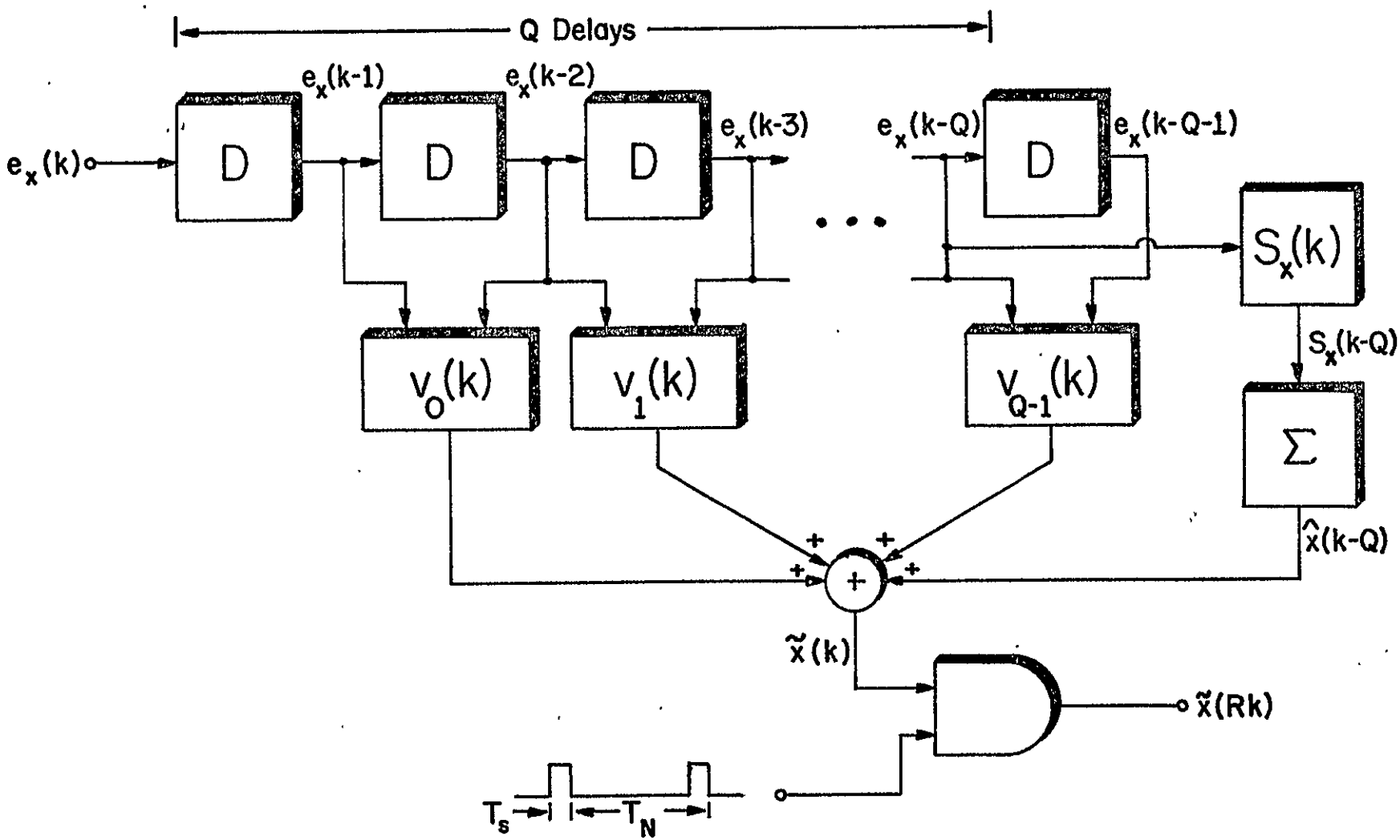


Fig. 3.3-4. Modified Realization of ADM to PCM Converter with Non-Recursive Filter

error that could arise is one minimum step size.

3.3.1 Realizing an Ideal Digital LPF

Now that we have derived a digital structure for ADM to PCM conversion, the only design that remains is the choice of the filter coefficients, g(j). Since the objective of the filter is to eliminate the spurious out-of-band frequency components, we wish to obtain the best out-of-band noise rejection with the least in-band signal distortion. In an attempt to achieve this, we employ a time domain design technique utilizing the unit step response of an ideal low pass filter (ILPF), i.e.,

$$g(t) = 0.5 + (1/\pi)\mathrm{Si}(2\pi f_c t - K_o) \tag{3.3-14}$$

where

$$\mathrm{Si}(\alpha) \equiv \int_0^{\alpha} \frac{\sin x}{x} dx \ , \tag{3.3-15}$$

$$f_c = f_m = f_N/2 = \text{the cutoff frequency of the ILPF} \ ,$$

$$K_o = 2\pi f_c T_d \tag{3.3-16}$$

and

$$T_d = \text{the constant time delay of the ILPF.}$$

In Fig. 3.3-5, we plot g(t) as a function of time normalized by $1/2\pi f_c$ when $K_o = 0$.

Although an ILPF is non-causal and, therefore, not physically realizable, this does not hinder our design of the digital filter. When a non-recursive digital filter is

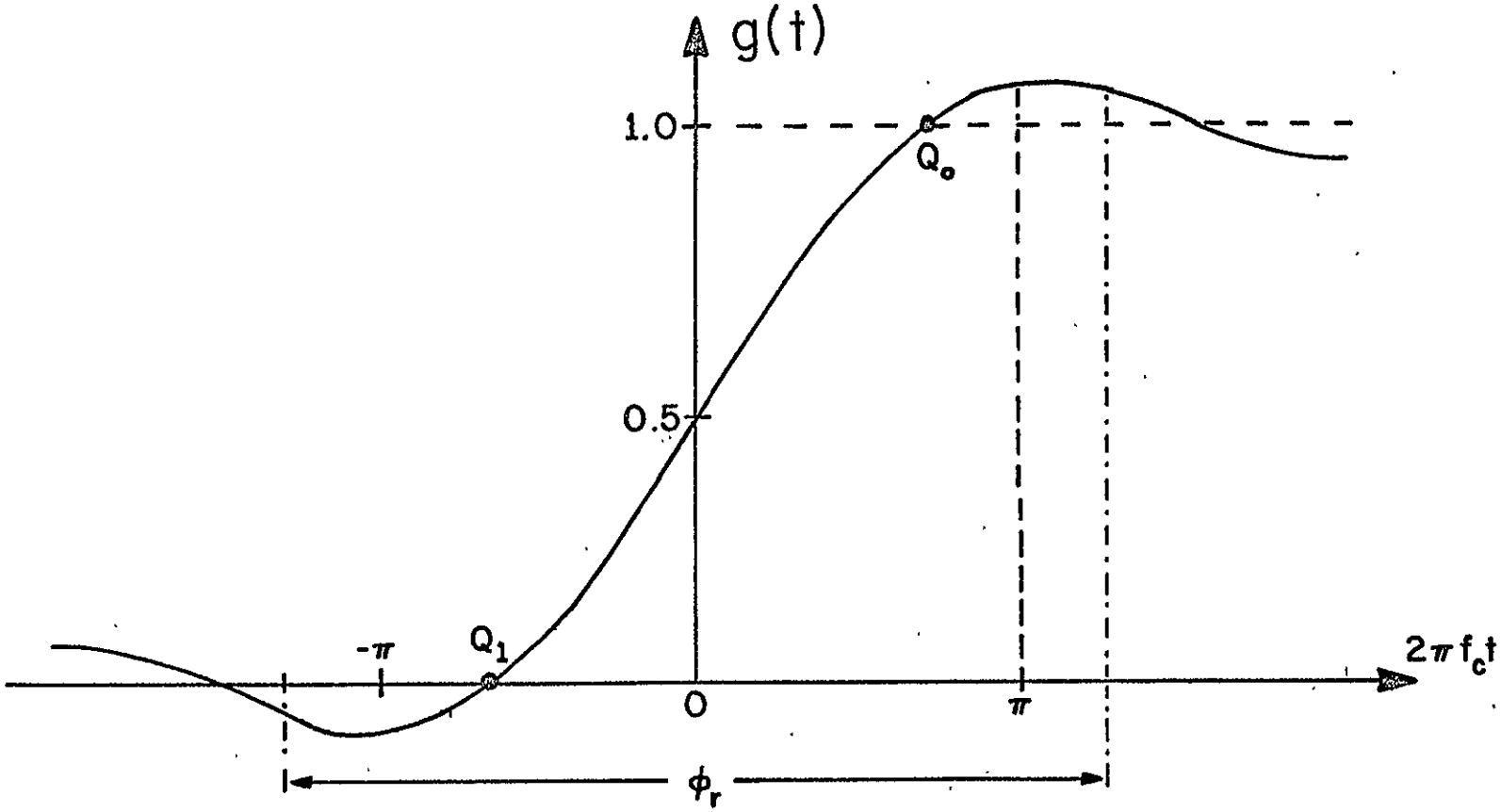


Fig. 3.3-5. Unit Step Response of an ILPF with Zero Time Delay

constructed, we can choose any coefficients desired to simulate a given characteristic. In our design, we are not concerned with the time delay, T_d, nor are we concerned with the anticipatory response of this analog ILPF. We only focus on the Q values of g(t) found between the asymptotic limits of 0 and 1 that g(t) approaches as t goes to minus and plus infinity. These Q values of g(t) complete the design.

Since we are not concerned with the ILPF time delay, we shall set K_o such that the values of the filter coefficients are symmetrically distributed about 0.5. To determine the value of Q, we must specify the desired filter rise time, T_r, or the region ϕ_r, shown in Fig. 3.3-5, outside of which we assume that g(j) takes on only values arbitrarily close to 0 or 1. If we set a value of ϕ_r, then we can calculate

$$T_r = \phi_r/2\pi f_c \,. \tag{3.3-17}$$

We observe that Q values of g(j) imply Q - 1 intervals of T_s seconds in T_r. Therefore,

$$T_r = (Q - 1)T_s \tag{3.3-18}$$

or

$$Q = \phi_r R/\pi + 1 \tag{3.3-19}$$

which is found by setting $R = f_s/f_N$. Because Q must be an integer, we shall specify it as

$$Q = \lceil \phi_r R/\pi \rceil + 1 \tag{3.3-20}$$

where

$\lceil \alpha \rceil \equiv$ the greatest integer $\leq \alpha$.

Now we can determine the filter coefficients by evaluating $g(t)$, at $t = jT_s$, symmetrically about 0.5. The coefficients are given as

$$g(j) = 0.5 + (1/\pi)\mathrm{Si}(\pi(j - R_o)/R) \tag{3.3-21}$$

where

$$j = 0, 1, \cdots, Q - 1,$$

and

$$R_o = 0.5\lceil \phi_r R/\pi \rceil . \tag{3.3-22}$$

Implicit in this evaluation of coefficients is that $g(j)$ is set to zero to the left of ϕ_r and set to one to its right. It is seen from Table 3.3-1 that the coefficients obtained from this technique, via Eq. (3.3-14), are within 1% of the values obtained by Goodman [21] for the example documented in this reference. We stress that these coefficients are independent of the input signal statistics.

3.3.2 Characteristics of the Ideal Digital LPF

The algorithm for the improved ADM estimate, i.e., Eq. (3.3-8), does not show the effect of the non-recursive filter on the original estimate, $\hat{x}(k)$. However, with some algebraic manipulations, we can transform the ADM to PCM converter shown in Fig. 3.3-2 into the cascade arrangement depicted in Fig. 3.3-6. Repeating Eq. (3.3-8),

j	g(j) OBTAINED BY ILPF TECHNIQUE	g(j) OBTAINED BY MINIMUM MEAN-SQUARE ERROR TECHNIQUE
1	0.15839	0.15996
2	0.37776	0.38571
3	0.62224	0.61437
4	0.84161	0.84013
5	1.00000	1.00009

TABLE 3.1-1. Coefficients for the Non-Recursive Digital LPF

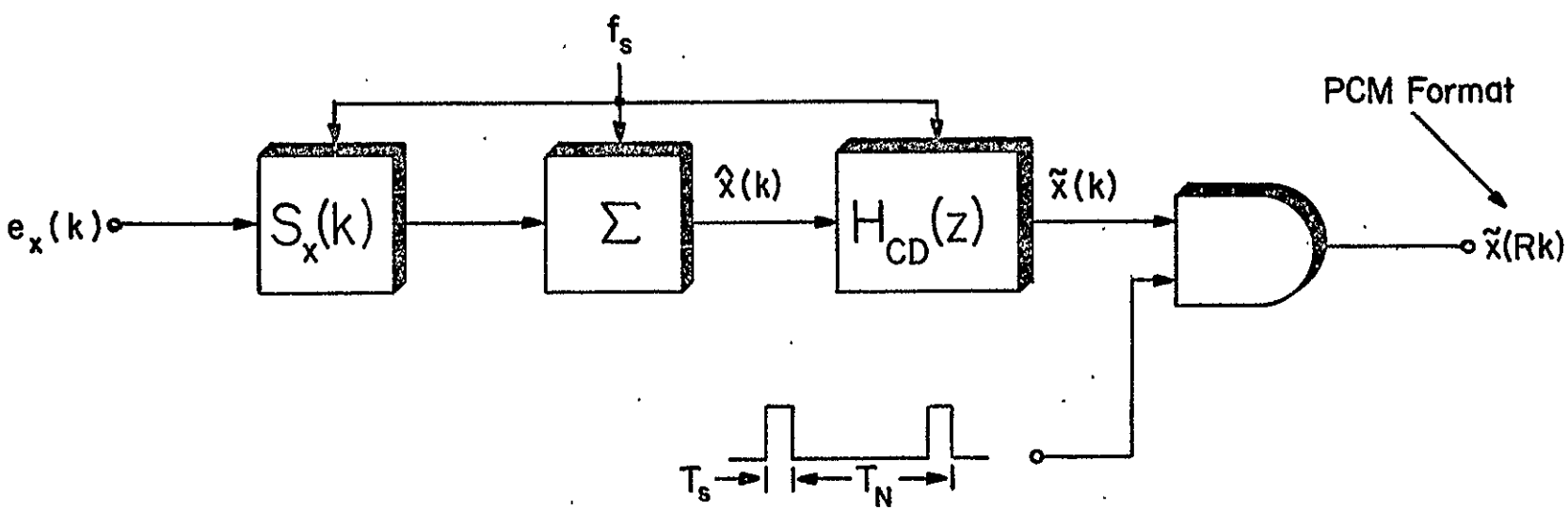


Fig. 3.3-6. Cascade Arrangement of the ADM to PCM Converter

$$\tilde{x}(k) = \hat{x}(k - Q) + \sum_{i=0}^{Q-1} g(i) S_x(k - i) \ , \qquad (3.3\text{-}8)$$

and observing that Eq. (2.1-4) can be rewritten as

$$S_x(k) = \hat{x}(k) - \hat{x}(k - 1) \ , \qquad (2.1\text{-}4)$$

we now see that

$$\tilde{x}(k) = \hat{x}(k - Q) + \sum_{i=0}^{Q-1} g(i) \cdot [\hat{x}(k - 1) - \hat{x}(k - i - 1)] \ . \qquad (3.3\text{-}23)$$

Taking the Z-transform of Eq. (3.3-23), assuming zero initial conditions, we obtain the form of the cascade filter,

$$H_{CD}(z) = \tilde{X}(z)/\hat{X}(z) = z^{-Q} + (1 - z^{-1}) \sum_{i=0}^{Q-1} g(i) z^{-1} \ , \qquad (3.3\text{-}24)$$

where

$\tilde{X}(z) \equiv$ the Z-transform of $\tilde{x}(k)$

and

$\hat{X}(z) \equiv$ the Z-transform of $\hat{x}(k)$.

By taking advantage of the symmetry of the g(i) coefficients about 0.5 and letting

$$z = \exp(j\omega T_s) \ , \qquad (3.3\text{-}25)$$

we can determine the frequency characteristics of the cascade filter. The following transfer function was derived from the filter structure used in simulation where Q was an even integer and g(0) = 0,

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

$$H_{CD}(f) = \exp(-j(Q+1)\pi f/2Rf_c)\Big(\{\cos(\pi(Q-1)f/2Rf_c)$$

$$+ 4\sin(\pi f/2Rf_c)[\sum_{i=1}^{Q/2-1} g'(Q/2-i)\sin(i\pi f/Rf_c)]\}$$

$$+ j\{-\sin(\pi(Q-1)f/2Rf_c) + 2\sin(\pi f/2Rf_c) \tag{3.3-26}$$

$$\cdot [0.5 + \sum_{i=1}^{Q/2-1} \cos(i\pi f/Rf_c)]\}\Big) ,$$

where

$$g'(i) = 0.5 - g(i) . \tag{3.3-27}$$

A derivation of $H_{CD}(f)$ is given in App. 3.

Since our simulation uses an input tone of frequency f_o and a cutoff frequency of $4f_o$ to determine SNR for the audio mode ADM, we shall plot $H_{CD}(f)$ on a frequency scale normalized to f_o. Figure 3.3-7 displays the amplitude and phase characteristics for the case where R = 8 and Q = 10, representing 9 values of g(t), between g(1) = 0.064 and g(9) = 0.936, symmetric about g(5) = 0.5. In this example, ϕ_r only extends from Q_0 to Q_1 (see Fig. 3.3-5) since larger values of ϕ_r, and consequently more g(i) coefficients, afford negligible SNR improvement.

In Fig. 3.3-8, we plot the amplitude characteristics of $H_{CD}(f)$ for the other two cases considered in our simulations, i.e., R = 6, Q = 8 and R = 4, Q = 6. In both cases, g(0) = 0 and the g(i)s fall between Q_0 and Q_1 on Fig. 3.3-5. Note the similarity with the amplitude characteristics when R = 8 and Q = 10. The phase characteristics for these last

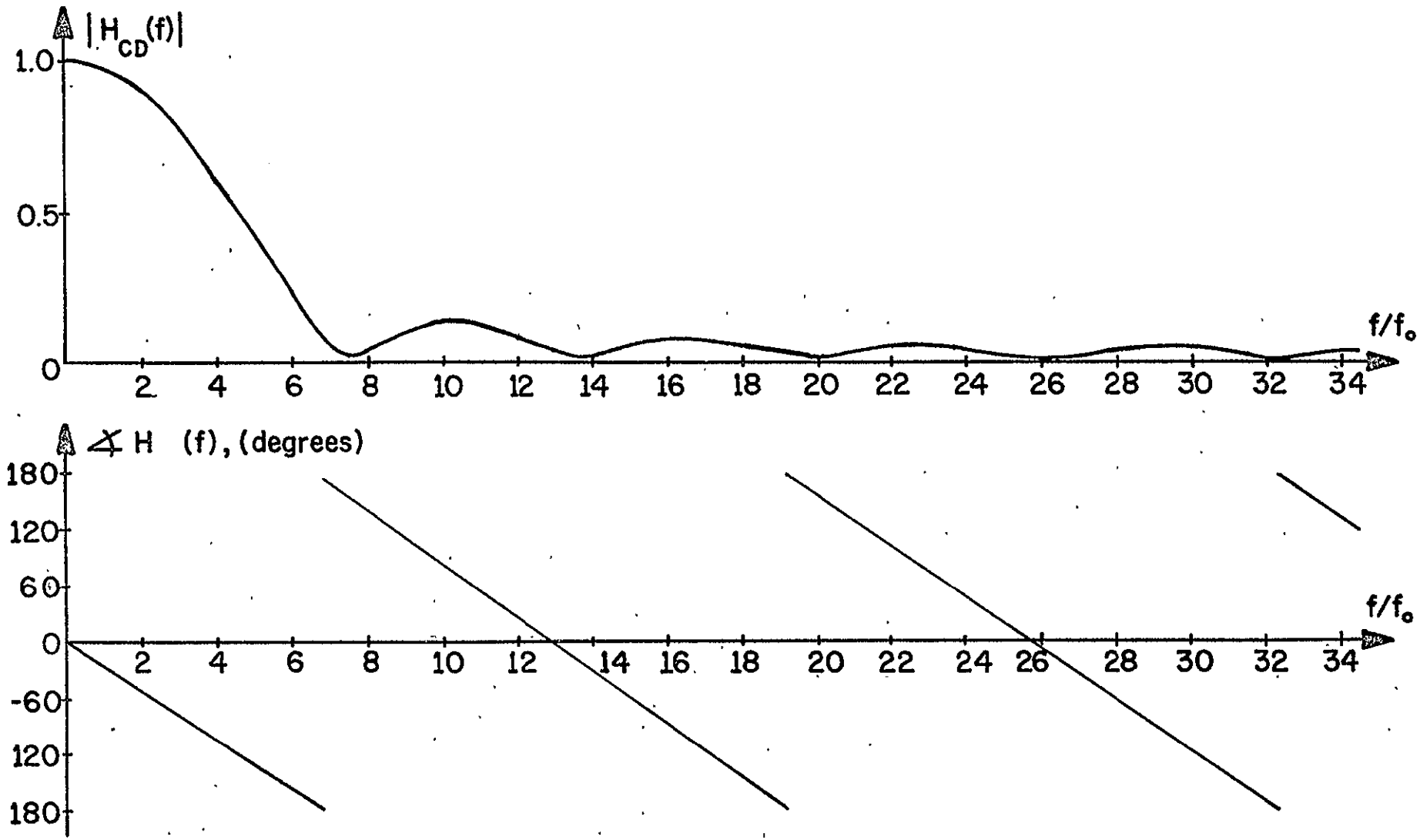


Fig. 3.3-7. Amplitude and Phase Characteristics of Converter Filter where R=8, Q=10

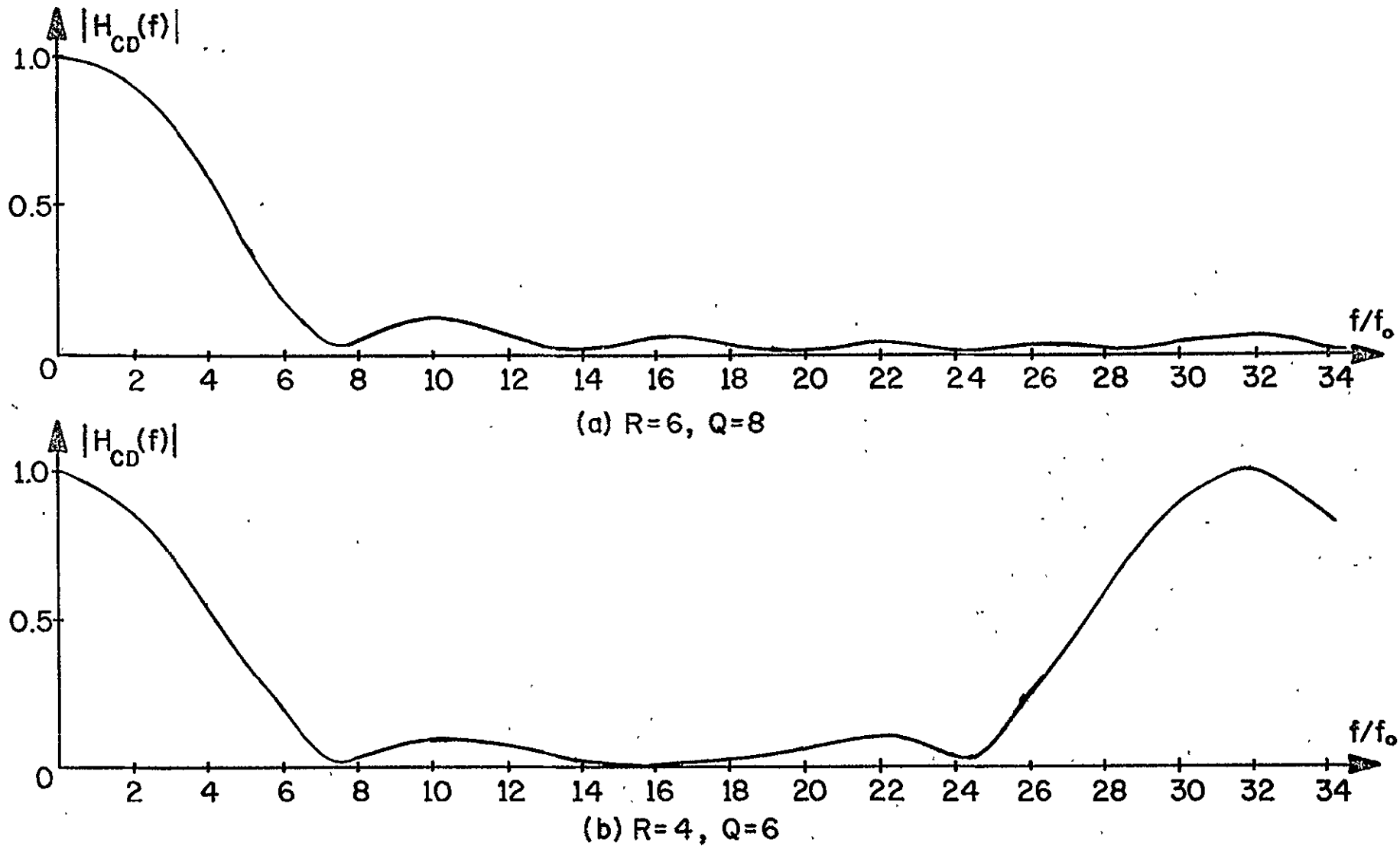


Fig.3.3-8. Amplitude Characteristics of Converter Filter where (a) R=6, Q=8, (b) R=4, Q=6

two cases are similar to the plot shown in Fig. 3.3-7. The phase is extremely linear with the slope changing from $-28f_o$ degrees/second when R = 8, Q = 10 to $-30f_o$ degrees/second and $-34f_o$ degrees/second when R = 6, Q = 8 and R = 4, Q = 6, respectively.

From the frequency characteristics of the converter filter, we see a disadvantage to the ADM to PCM converter described above. There is a definite shaping of the in-band spectrum due to the attenuation below f_c. This will cause distortion of the signal. In determining the performance curves, this attenuation was corrected for so as not to yield misleading results. If this was not done, the SNR of the converter with non-recursive filter would sometimes be greater than the SNR of the optimum converter (described in a later section), which employs analog demodulation. After the performance curves are presented, we shall discuss remedies for, and alternatives to, the in-band spectrum shaping.

Equally important as realizing that spectral shaping does occur is the explanation of why it exists at all in this system. Our initial design employs the unit step response of an <u>ideal</u> LPF simply because it has the most desirable frequency characteristics. However, the brick wall frequency characteristics only hold if we realize the <u>analog</u> ILPF. The ADM to PCM converter utilizes a digital LPF which uses a finite number of discrete points on the unit step response. $H_{CD}(f)$ will not approach a brick wall characteristic if we use more points on the tails of the g(t) curve.

This can be accomplished by choosing the points closer together. However, to maintain the same cutoff frequency, we would have to increase the ADM bit rate. But, in ADM to PCM conversion of this type, we are constrained to a previously set ADM rate. Thus, we are not at liberty to use this technique to eliminate spectral shaping.

3.3.3 Converter Simulation and Sinusoidal Response

Using our PDP 8/L computer with 8K memory we simulated the Song audio mode ADM to encode test signals. Then we simulated the ADM to PCM converter with non-recursive filter exactly as depicted in Fig. 3.3-2. The original system used $R = 8$, $Q = 10$ and thus had 9 filter coefficients. These coefficients were approximated by the nearest integral multiple of 1/16 to simulate a hard-wired scaler. The initial test employed a sinusoidal signal encoded by the ADM operating at a rate which was an integral multiple of the input sinusoid frequency. The ADM bit stream, $\{e_x(k)\}$, was then used as the input to the converter.

Whenever the ADM bit rate is an integral multiple of the input sinusoid frequency, we know that the estimate, $\hat{x}(k)$, will assume a periodic sinusoidal steady state pattern. Since the improved estimate, $\tilde{x}(k)$, and the resulting PCM samples are derived from $\hat{x}(k)$ by linear filtering operations, these signals also take on a periodic pattern. The periodicity of $\hat{x}(k)$ and $\tilde{x}(k)$ was verified by the system simulation.

To show the qualitative improvement of the ADM estimate afforded by the non-recursive digital filter, we have displayed, in Fig. 3.3-9, the improved ADM estimate, $\tilde{x}(k)$. We have also plotted the delayed analog input and the original ADM estimate, $\hat{x}(k)$. These curves represent a converter where R = 8 and Q = 10, exactly as described in Sec. 3.3.2. This emphatically shows the "poor" PCM samples that can be obtained from $\hat{x}(k)$ and the consistently good quality of virtually all the samples of $\tilde{x}(k)$.

3.3.4 Evaluation of Performance

The families of performance curves are obtained with the same approach as discussed in Sec. 2.8. With a single tone input, we obtain a periodic pattern for $\hat{x}(k)$, $\tilde{x}(k)$ and the resulting PCM samples. Thus, all of these waveforms can be expressed as a Fourier series. The frequency components of these series can be calculated and used to determine output SNR. Let the frequency of the sinusoid be f_o. We have shown in Ch. 2, adhering to the assumptions made there, that the ADM estimate, and any signal derived from it, can be expressed in the form

$$x(t) = C_o + \sum_{n=1}^{\infty} C_n \cos(2\pi n f_o t + \phi_n) \ , \qquad (3.3\text{-}28)$$

where C_o, C_n and ϕ_n are defined as the standard Fourier amplitude and phase coefficients, i.e., Eqs. (2.8-3b) through (2.8-3f). We have shown, in Sec. 2.8, that the calculation of these coefficients is readily adaptable to computer

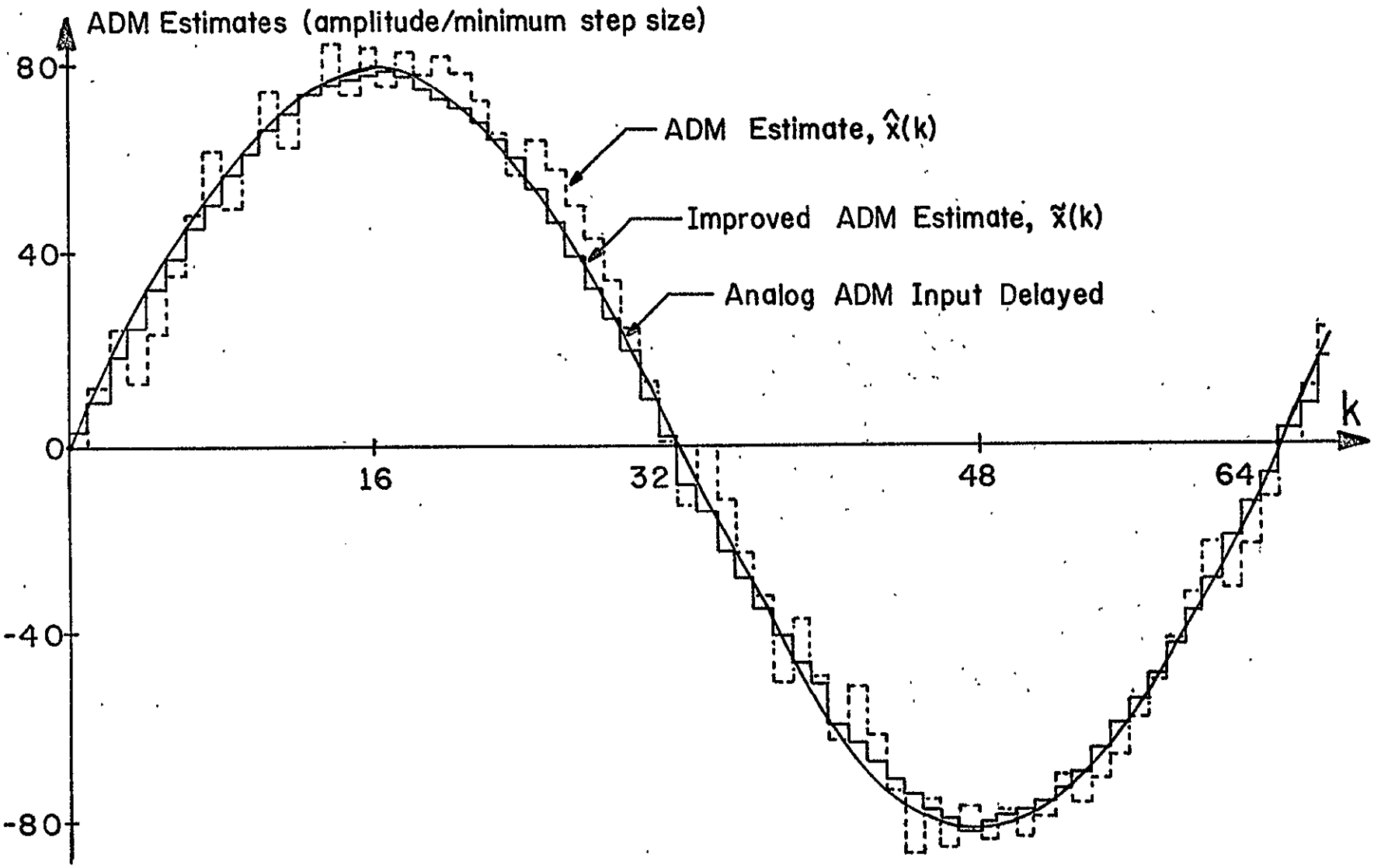


Fig. 3.3-9. ADM Estimate: Before and After Non-Recursive Filter

simulation.

In determining the frequency spectrum of the PCM signal, we utilize the Fourier series technique in two different ways, depending on the conversion system under analysis. The time domain approach simulates the entire converter and determines the Fourier components, C_n, from the periodic pattern of the PCM samples. The time and frequency domain approach computes the amplitude and phase components of the ADM estimate (C_n and ϕ_n) and then simulates digital low pass filtering, resampling and holding with the appropriate transfer functions in the frequency domain. Digital low pass filtering uses $H_{CD}(f)$ from Eq. (3.3-26); resampling employs the frequency shifting given in Eq. (3.2-3); and the holding circuit, for the Nyquist period, is described by [22]

$$G_H(f) = \frac{\sin(\pi f/f_N)}{\pi f/f_N} . \qquad (3.3\text{-}29)$$

The former of these two techniques cannot be used if we wish to correct for the spectral shaping caused by $H_{CD}(f)$. Consequently, we must employ the latter method. Applying this approach to the cascade arrangement of the ADM to PCM converter with the non-recursive filter (Fig. 3.3-6), the held PCM spectrum is given as

$$\tilde{X}_{NH}(f) = G_H(f) \sum_{i=-\infty}^{\infty} \hat{X}(f + if_N) H'_{CD}(f + if_N) , \qquad (3.3\text{-}30)$$

where

$\hat{X}(f) \equiv$ the frequency spectrum of the ADM estimate

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

and $H'_{CD}(f)$ represents the cascade digital LPF after spectral shaping has been eliminated. The correction is accomplished during simulation by setting the amplitude of $H_{CD}(f)$ to be 1 at $f = f_o$, $2f_o$ and $3f_o$, and by specifying it as 0.707 at $f = f_c = 4f_o$. The phase of $H_{CD}(f)$ was left unchanged. If this correction was not made, $H_{CD}(f)$ would attenuate the second through fourth harmonics. We would then have less harmonic noise and, therefore, a higher SNR. All operations in Eq. (3.3-30) are performed on complex quantities. Thus, the multiplications and additions are actually vector products and vector sums.

Since we are concerned with an audio mode ADM, we again choose an output LPF, to return to the analog domain, which is applicable to voice signals, i.e., a fourth-order Butterworth LPF whose frequency characteristics are given in Eq. (2.8-7). Exactly as in Sec. 2.8.2, we choose the LPF cutoff frequency, f_c, to be $4f_o$ where f_o is our input tone. The LPF attenuation factor, α_n, is the same as Eq. (2.8-9) and the output signal-to-noise ratio, SNR_o, is given by Eq. (2.8-12), where C_n now represent the strength of the Fourier components of the output PCM signal obtained from Eq. (3.3-30).

To avoid several infinite sums and products in the actual calculation of the held PCM spectrum, $\hat{X}_{NH}(f)$, we truncated the spectrum of $\tilde{x}(k)$, i.e., $\hat{X}(f)H'_{CD}(f)$, after the ninth harmonic. The attenuation afforded by the digital LPF causes the frequency components of $\tilde{x}(k)$ to be insignificant

from the tenth harmonic on. It was then possible to reduce the infinite sum in Eq. (3.3-30) to a sum from -2 to +2. Because of the truncation of the spectrum of $\tilde{x}(k)$, further sliding along the frequency axis added nothing significant to $\tilde{X}_{NH}(f)$, especially after the attenuating effect of the holding circuit, $G_H(f)$. Finally, in determining SNR_o we truncated the noise power after the ninth harmonic since $(\alpha_{10}C_{10})^2$ was negligible in comparison to the noise due to the second through ninth harmonics. As in Ch. 2, we found that, for the same input signal amplitude, the periodic pattern that the ADM estimate assumes and, consequently, the SNR_o are very dependent upon the starting point of the input sinusoid. To obtain a truly representative value of SNR_o, we averaged over 32 different starting points. All SNR curves are given at the end of this chapter to facilitate comparison between the various conversion techniques discussed.

3.4 Other Digital Conversion Techniques

In addition to the ADM estimate converter and the converter with non-recursive filter discussed above, there are several other digital techniques that can be applied to achieve ADM to PCM conversion. The most straightforward approach is to use a cascade digital LPF before we resample. We could accumulate the step sizes and then LPF the estimate, or, low pass filter $S_x(k)$ and then accumulate the filtered step sizes. Both systems are equivalent because the LPF and

the accumulator are linear devices and, therefore, interchangeable.

Although L.D.J. Eggermont [23] proposed an ADM to PCM converter following Goodman's design technique [24], he actually used a non-recursive digital filter in cascade, before the accumulator. He reported that "at a sampling frequency of 64K Hz the system performed very well for speech encoding" [25]. However, no performance curves were given for his ADM to PCM converter. We have considered using this same cascade technique, except with a recursive digital LPF before the accumulator. A recursive digital LPF can achieve a very sharp frequency cutoff with a minimum amount of hardware. We have designed second- through fourth-order recursive LPFs using both the impulse invariant method and the squared-magnitude method [26]. However, with a recursive digital filter, the accuracy of the filter coefficients is critical in maintaining stability and in obtaining the desired frequency characteristics. Since we must increase the hardware complexity to guarantee accurate coefficients and because the converter derived in Sec. 3.3 performs so well, these ADM to PCM converters, with recursive filter, were never simulated. As long as the filter has a sharp frequency cutoff, this type of converter will perform well; but, we must deal with the coefficient problem mentioned above.

Several other authors have very recently approached the problem of DM to PCM conversion. T. Ohno, H. Kuwahara,

M. Miyata and K. Imai [27] use a linear DM to achieve analog to PCM conversion. Since they must operate the linear DM at a very high bit rate (8.192M Hz), their system converts to PCM by using two cascaded bit reduction devices. They do not consider adaptive DM processors at all. On the other hand, J. H. Miller [28] deals with ADM to PCM conversion using only digital hardware. However, he has the added problem that the ADM encoder which he must use is completely analog. Thus, he must derive the digital equivalents to all the analog circuits. L. B. Jackson, J. F. Kaiser and H. S. McDonald [29] also considered the use of a linear DM in converting from analog to PCM encoded signals. Their scheme employs a digital differential technique before a first-order digital LPF converts to PCM format. None of these, however, add any additional concepts to the actual problem of converting from ADM encoded signals to PCM type signals. The important thread that is common to all conversion techniques is the absolute need for some type of low pass filtering before resampling to avoid the disastrous aliasing effect discussed in Sec. 3.2.

3.5 Analog Demodulation Technique

The last ADM to PCM system that we consider decodes $e_x(k)$ to produce the ADM estimate and then uses an analog LPF to demodulate $\hat{x}(k)$ before resampling at the Nyquist rate. This analog demodulation converter is shown in Fig. 3.5-1. We purposely omit the quantizer and PCM encoder after the

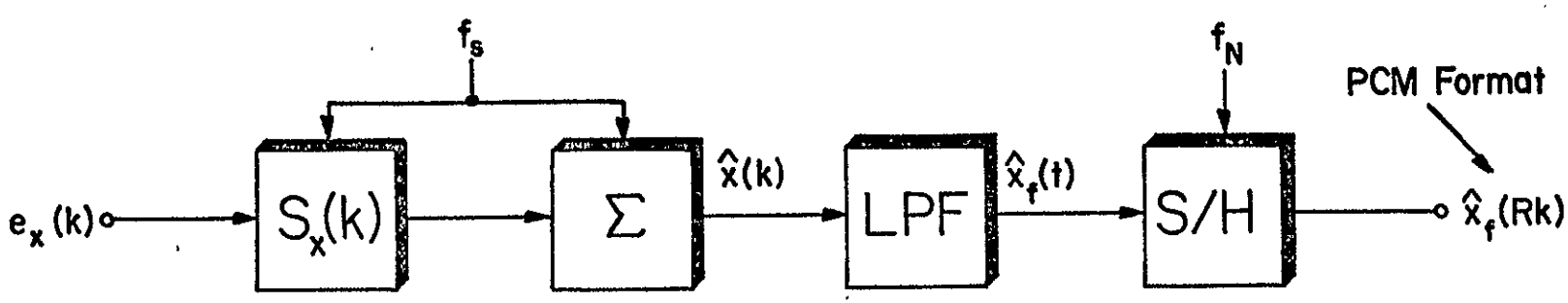


Fig. 3.5-1. Analog Demodulation Converter

sample-and-hold (S/H) circuit because our performance evaluation is based on harmonic noise and the quantization noise is considered negligible.

Although this is not a completely digital system and we do not want to construct this device, it is the optimum converter if an <u>ideal</u> LPF is used to demodulate $\hat{x}(k)$. It will yield the best performance, i.e., highest SNR_o, because the ILPF eliminates all the out-of-band frequency components from the ADM estimate before PCM sampling. Thus, there is no aliasing to degrade the PCM signal. Analyzing this system gives a good basis of comparison for the performance curves presented later.

The ideal analog demodulation converter has been simulated and its performance determined by using the time and frequency domain approach. The ADM encoder is simulated in the time domain and we calculate the Fourier spectrum of $\hat{x}(k)$ for a single tone input using the technique described in Sec. 2.8. The ILPF is simulated in the frequency domain by eliminating all frequency components of $\hat{x}(k)$ above f_c and letting all harmonics below f_c pass unattenuated. The S/H circuit is simulated via the frequency shifting given in Eq. (3.2-3) and the holding transfer function, $G_H(f)$, given by Eq. (3.3-29).

Let us define the transfer function of the ILPF to be

$$\begin{aligned} G_I(f) &= 1, & f < f_c\ , \\ &= 0.707, & f = f_c\ , \\ &= 0, & f > f_c\ . \end{aligned} \tag{3.5-1}$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

The output of the ideal analog demodulation converter is denoted as $\hat{x}_f(Rk)$ and its frequency spectrum can be expressed in the following manner:

$$\hat{X}_{fNH}(f) = G_H(f) \sum_{i=-\infty}^{\infty} \hat{X}(f + if_N) G_I(f + if_N) \quad . \qquad (3.5\text{-}2)$$

We must define $G_I(f)$ at f_c because, as before, the cutoff frequency is set to four times the test tone and $\hat{x}(k)$ will have a harmonic at f_c. In calculating SNR_o, we again use the fourth-order Butterworth filter, the expression for SNR_o given in Eq. (2.8-12) and, as before, truncate the noise power after the ninth harmonic. Similarly, SNR_o acts as a random variable dependent upon the starting point of the input sinusoid and we average over 32 different starting points to obtain a mean SNR_o.

Before we actually calculate SNR_o for the ideal analog demodulation converter, we can estimate that its value will be very close to the SNR of the ADM estimate, $\hat{x}(k)$. Since we have assumed impulse sampling at f_N and negligible quantization effect, we can conclude that both $\hat{x}_f(Rk)$ and $\hat{x}(k)$ have exactly the same in-band frequency spectrum. They both have out-of-band harmonics which are attenuated by the output LPF and, therefore, have a secondary effect on the SNR. Thus, the SNR of $\hat{x}(k)$ will be a very close approximation to the SNR of $\hat{x}_f(Rk)$.

3.6 Comparison of Conversion Systems

The three converters discussed at length in this chapter were completely simulated and tested: the ADM estimate converter, shown in Fig. 3.1-1, the ADM to PCM converter with non-recursive filter, given in Fig. 3.3-2, and the ideal analog demodulation converter, depicted in Fig. 3.5-1. In this section we present the technique used to evaluate the SNR for each system and its corresponding family of performance curves. These curves display the variation of SNR_o with relative input signal power for several ADM bit rates. Thus, we are able to present an evaluation of the performance of all ADM to PCM conversion systems.

3.6.1 SNR Evaluation

To determine the Fourier components of the held PCM samples, the time domain approach was used for the ADM estimate converter. The time and frequency domain approach was used for the ADM to PCM converter with non-recursive filter and for the ideal analog demodulation converter. In all cases, we used a fourth-order Butterworth LPF before calculating SNR_o. In determining SNR_o, we truncated the noise power after the ninth harmonic.

When evaluating the performance of the first converter, we found that for the same input and ADM bit rate, the position of the gating pulse in the Nyquist interval would give rise to different values of SNR_o. In Fig. 3.1-1, we show how the gating pulse, of duration T_s seconds, produces $\hat{x}(Rk)$

by permitting one value of $\hat{x}(k)$ through the AND gate every T_N seconds. The gating time can be expressed as

$$t_g = kT_N + jT_s \ , \qquad (3.6\text{-}1)$$

where

$$j = 0, 1, 2, \cdots, R - 1 \ .$$

Initially, it seemed surprising that even though the sinusoidal steady state pattern of $\hat{x}(k)$ did not change, we could still obtain different values of SNR_o by varying the gating time, i.e., j in Eq. (3.6-1). This variation of SNR_o can best be explained by viewing the gating times (kT_N) as fixed values and allowing the signal $\hat{x}(k)$ to be shifted by jT_s, instead of shifting the gating times for a fixed $\hat{x}(k)$.

Now we see that we have a fixed time reference and the phase, or time delay, of the sinusoidal steady state pattern of $\hat{x}(k)$ varies with jT_s. Therefore, although the shape of $\hat{x}(k)$ and, consequently, its amplitude spectrum remain the same for all values of j, the phase spectrum does not. By viewing the gating as resampling which causes aliasing, we now realize that the overlapping spectra, when added vectorially, yield different results depending upon the phase spectrum of $\hat{x}(k)$. Therefore, SNR_o will be different for different values of jT_s.

For all converters, we found that SNR_o acted as a random variable dependent upon the starting point of the input sinusoid. For the first converter, both the gating time and the starting point were varied to yield 32 values of SNR_o.

When evaluating the last two converters only the starting point was changed 32 times because the time and frequency domain approach was used and it does not use a gating circuit. For all cases, the mean SNR_o was calculated along with the standard deviation of SNR_o.

3.6.2 Three Families of SNR Curves

In Fig. 3.6-1 we show, for the three ADM to PCM converters, the output signal-to-noise ratio, SNR_o, in dB versus relative input signal power over a range of 54 dB for ratios of $f_s/f_N = R = 8$, 6 and 4. As noted in Ch. 2, the input signal amplitude was varied from 5S, corresponding to -6 dB, to 1280S at 42 dB to generate the curves. Of course, S is the minimum ADM step size found in the Song audio mode algorithm. For these performance curves, the test tone frequency was f_o; the LPF cutoff frequency is set at $f_c = 4f_o$; and the PCM rate employed is $f_N = 2f_c = 8f_o$. The plots shown are smooth curves drawn through windows of ±1 standard deviation about the mean SNR_o for various input amplitudes. For these curves, the standard deviation was approximately 1-2 dB. The ADM to PCM converter with non-recursive filter produces an 8-10 dB improvement over the ADM estimate converter and, for moderately high ADM bit rates, it comes within 1-2 dB of the ideal analog demodulation converter, which is the optimal system.

We see from the performance curves that the ADM to PCM converter with non-recursive filter yields a good represen-

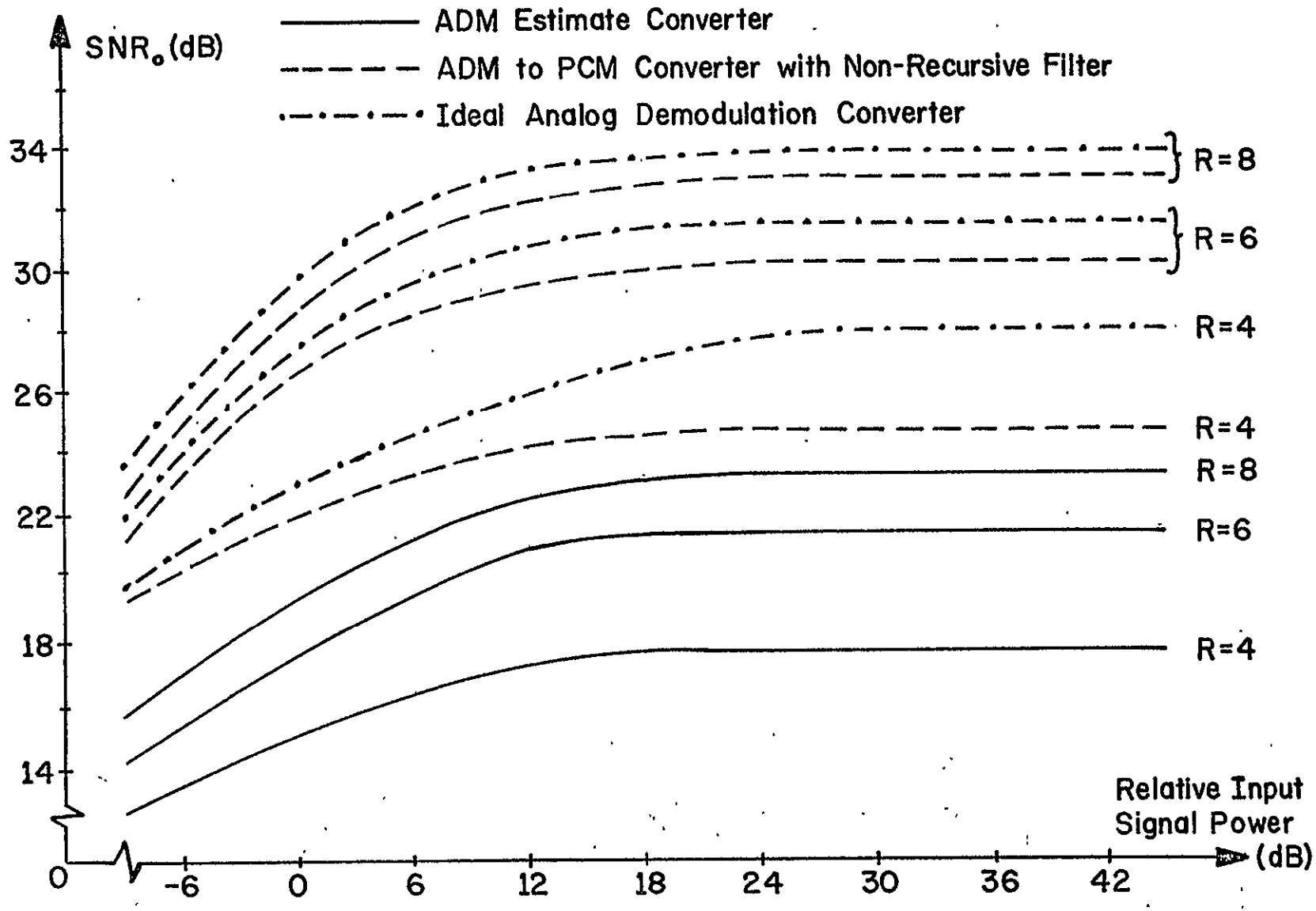


Fig. 3.6-1. Performance Curves for ADM to PCM Converters

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

tation of the original signal in PCM format: By examining its hardware structure in Fig. 3.3-2, we observe that it is easily realizable with readily available digital circuits. In our simulation, we have used only 9, 7 and 5 filter coefficients for R = 8, 6 and 4, respectively. It has already been noted that more coefficients resulted in negligible increase in SNR_o. The other advantage of this structure is that the accuracy of the filter coefficients is not extremely critical. The coefficients used in the simulations were approximated by the nearest integral multiple of 1/16. This represents, at most, an additional 4 bits needed in the internal arithmetic. For this case, we suffered about 0.1 dB decrease in SNR_o.

Let us return to the problem of spectral shaping caused by the frequency characteristics of the converter filter. It has already been noted that we are not at liberty to rectify this by choosing more filter coefficients closer together on the g(t) curve. The spectral shaping can, however, be remedied by using an output digital filter that has in-band characteristics which are the inverse of $H_{CD}(f)$. Alternatively, it could be corrected by using a better method of choosing the filter coefficients, such as starting with the frequency domain filter characteristics and then calculating the time domain coefficients.

A final idea would be to redesign the system, similar to Fig. 3.3-6, and construct a purely cascaded digital filter. However, by using the rather gross time domain design

method presented in Sec. 3.3, we have come to within 1-2 dB of the optimum converter performance. Thus, one can question the need to resort to very sophisticated filter design techniques. We have, in any event, shown conclusively that we need some sort of LPF before PCM sampling and why we cannot do without it. Thus, regardless of the structure and how it is designed and implemented, an ADM to PCM converter must employ a LPF before resampling if we are to achieve acceptable performance.

CHAPTER 4

CONVERSION FROM PCM ENCODED SIGNALS TO ADM ENCODED SIGNALS

In this chapter, we consider the dual of the problem addressed in Ch. 3, that is, conversion from PCM to ADM format. We can describe a PCM to ADM converter as a device which operates on values of the information source, occurring at the Nyquist (or PCM) rate, and produces an ADM bit stream. Since there is a unique mapping between the ADM bit stream and the ADM estimate (when the ADM initial conditions are specified), and since the ADM operates at a rate several times higher than the Nyquist rate, we are converting from a low "information" rate (the PCM samples) to a high "information" rate (the ADM estimates).

Several techniques are developed to perform this type of conversion. One uses the probabalistic statistics of the information source. Another employs the spectral parameters of the input signal. Still another is completely non-parametric. All techniques deal with the lack of information about the signal excursion between PCM samples. The resolution of this problem dictates the structure of each PCM to ADM converter.

All the PCM to ADM converters that we design are restricted to circuits that are physically realizable with standard digital hardware and which entail a minimum amount of hard-

ware complexity. The sole constraint applied to all conversion systems is that the ADM bit rate is an integral multiple of the Nyquist rate. The PCM to ADM converters presented in this chapter wil either be statistically analyzed or simulated on a digital computer or both. In all cases, we shall generate SNR curves so that the performance of the converters can be objectively evaluated.

4.1 DM Signal Estimate Tree

To illustrate the conceptual difficulties associated with PCM to ADM conversion, we introduce a "DM signal estimate tree" and construct part of this tree for a particular variable step size DM. A "DM signal estimate tree" is a graph of all possible paths that the DM estimate, $\hat{x}(k)$, may follow starting with a set of initial conditions. The paths are generated from all possible binary sequences of the DM bits, $e_x(k)$. Every variable step size DM will give rise to its own particular "estimate tree" but each tree exhibits the same problems for PCM to ADM conversion. Thus, the conceptual difficulties are universal to all DMs.

4.1.1 Paths through the Signal Estimate Tree

Since we are primarily concerned with speech signals, without loss of generality, we shall invoke the Song audio mode algorithm whenever we refer to an ADM. In Fig. 4.1-1, we show part of a "DM estimate tree" for this particular step size algorithm, i.e., Eq. (2.1-6), on which we have super-

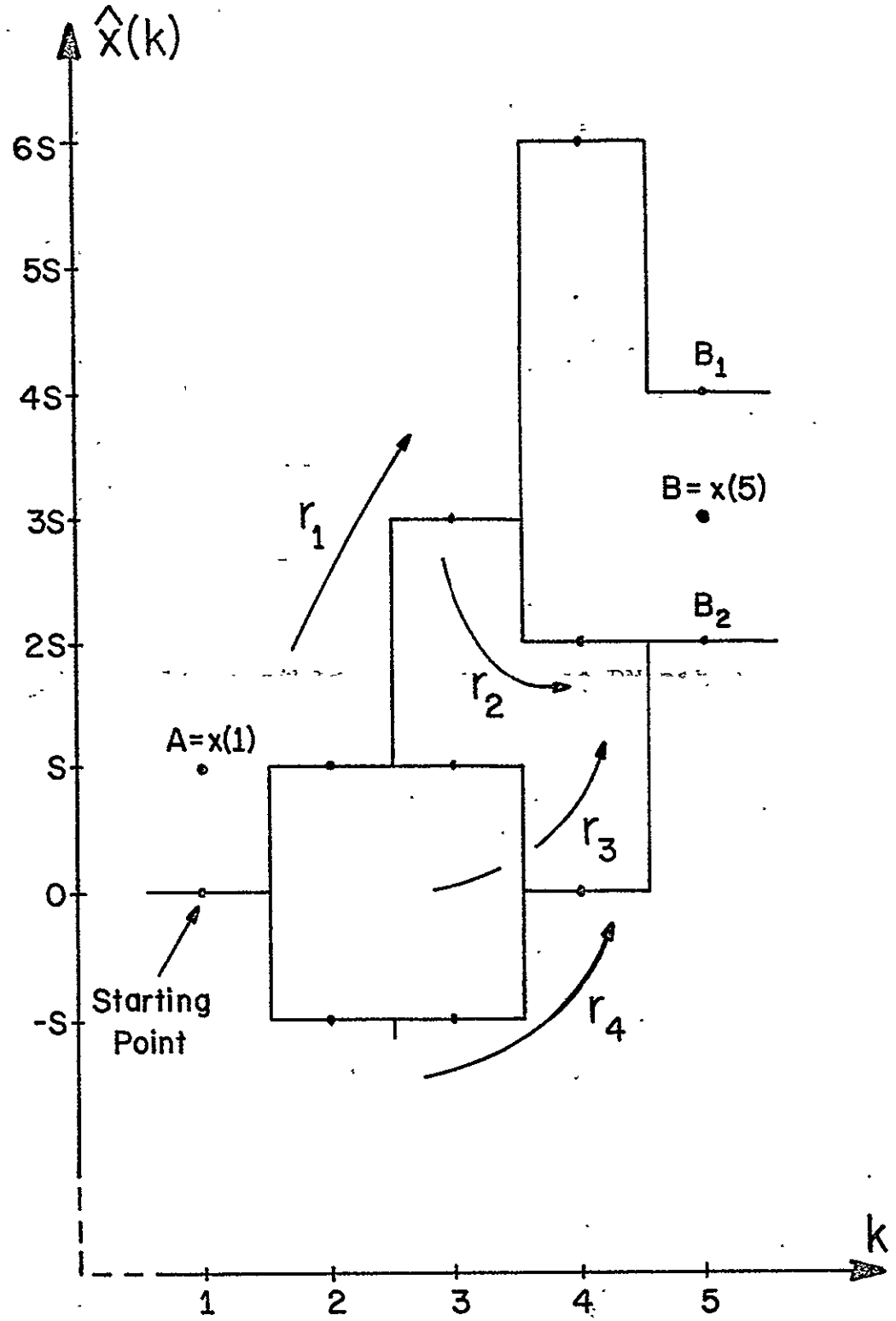


Fig. 4.1-1. Possible Paths Through an "ADM Signal Estimate Tree"

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

imposed two PCM samples, at points A and B. Now we observe where the difficulty lies in converting from PCM to ADM format. In going from the starting point to points on the "DM estimate tree" which are adjacent to point B, that is, points B_1 and B_2, we can traverse any of four possible paths: r_1, r_2, r_3 or r_4. Which path to choose is the problem in PCM to ADM conversion. We shall term this the "multipath" problem.

4.1.2 Path Endpoints

The complexity of the "multipath" problem can be somewhat reduced by coping with another conceptual difficulty, the "endpoint" problem. Because the "DM estimate tree" diverges so rapidly for a variable step size algorithm, the PCM samples may not always lie on a branch in the tree. Although a PCM sample, like point B, in Fig. 4.1-1, can take on any one of the integral values of S, the estimate, at that time, is restricted to endpoints of the ADM paths originating from the starting point. By first choosing endpoints for the PCM samples, we automatically eliminate some possible paths through the "DM estimate tree" and, thereby, reduce the complexity of the "multipath" problem and the resulting PCM to ADM conversion.

4.2 Statistical PCM-ADM Converter

There are several approaches to the PCM to ADM conversion problem, but they all require the addition or derivation of more information about the signal source or about the

excursion of the actual signal. For the techniques considered later, we shall estimate the signal excursion at discrete points between PCM samples and essentially fit these values to a path through the ADM signal estimate tree. For the statistical PCM to ADM conversion technique, we must introduce a set of probabalistic statistics for the input signal.

Let us assume that we have a stationary information source, $x(t)$, which is bandlimited to f_m, and that the ADM bit rate, f_s, is an integral multiple of the PCM rate, f_N; that is,

$$f_s = Rf_N \ , \tag{4.2-1}$$

where

R = a positive integer greater than 1,

and

$$f_N = 2f_m \ . \tag{4.2-2}$$

We also specify an R-dimension joint probability density function, representing R sampled values of $x(t)$ in the interval T_N, where

$$T_N = 1/f_N \ . \tag{4.2-3}$$

This can be expressed as

$$p_{\underline{x}}(\underline{\alpha}) = p_{\underline{x}}(\alpha_1, \alpha_2, \cdots, \alpha_R) \tag{4.2-4}$$

where

$\underline{x}$ represents the R-dimensional vector of $x(t)$.

We can assume that the conversion is a continuing process and we are choosing a path between each adjacent pair of PCM samples. Thus, the initial conditions for each tree are known from the previous path chosen. The estimate path is mapped into one of the 2^R possible R-bit sequences representing $e_x(k)$ in each Nyquist interval. When evaluating the "estimate tree" in each PCM period, we need not consider all 2^R sequences in our quest for the most likely path. As seen in Fig. 4.1-1, we can limit ourselves to those paths actually going between possible endpoints.

The method that we shall describe for determining the most likely path is extremely general and can be applied to virtually any ADM system. This technique is based on the most elementary principle of DM operation. If the signal value is greater than the estimate, i.e., $x(k) > \hat{x}(k)$, then the estimate must increase and $e_x(k) = +1$. Likewise, if the signal value is less than the estimate, i.e., $x(k) < \hat{x}(k)$, then the estimate must decrease and $e_x(k) = -1$. Using this principle, along with the joint statistics of $\underline{x}$, we can calculate the probability of each possible path. The obvious criterion to use is: the most likely path has the highest probability. That path is chosen and mapped into its ADM bit sequence. By using the given PCM values at the endpoints, we can eliminate some of the possible paths at the outset and thus reduce the necessary computation.

4.2.1 Choosing Tree Path Endpoints

For each Nyquist interval, we have a starting point defined as the end of the estimate path traversing the previous interval. Emanating from the starting point will be several possible estimate paths which terminate at or are adjacent to the next PCM sample. If no path goes through the next PCM sample, then we have two possible endpoints, as shown on the tree in Fig. 4.1-1. Since we wish to use the PCM samples as advantageously as possible, point B shall be used to determine the path endpoint.

The criterion that we shall employ to choose the endpoint is extremely simple: select the point that will yield the least error between it and the PCM sample. Let us define the error for points B_1 and B_2, respectively, to be

$$\varepsilon_{B_1} = |B_1 - B| \tag{4.2-5}$$

and

$$\varepsilon_{B_2} = |B_2 - B| \ . \tag{4.2-6}$$

We apply the following rule to choose the endpoint:

$$\begin{aligned} &\text{if} \quad \varepsilon_{B_1} < \varepsilon_{B_2} \ , \quad \text{choose } B_1 \ , \\ &\text{if} \quad \varepsilon_{B_2} < \varepsilon_{B_1} \ , \quad \text{choose } B_2 \ , \\ &\text{if} \quad \varepsilon_{B_1} = \varepsilon_{B_2} \ , \quad \text{choose } B_1 \text{ and } B_2 \ . \end{aligned} \tag{4.2-7}$$

This is equivalent to choosing the endpoint closest to the PCM sample. By choosing either B_1 or B_2, we automatically eliminate all paths going to B_2 or B_1.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

4.2.2 Choosing the Most Likely Path

To best explain the technique for determining the most likely path, we apply this procedure to the example given in Fig. 4.1-1. By using the initial conditions of the "estimate tree" and the PCM sample at the beginning of the path, we determine the first $e_x(k)$ and try to eliminate some possible paths. From Fig. 4.1-1, we see that $\varepsilon_{B_1} = \varepsilon_{B_2}$, so we choose both endpoint, B_1 or B_2, and must consider paths r_1, r_2, r_3 and r_4. Observe that the PCM sample $A = x(1) > \hat{x}(1)$ so that $e_x(1) = +1$ and path r_4 is eliminated. Now we must only calculate the probability of paths r_1, r_2 and r_3.

The probability density function used to determine the most likely path also takes advantage of the specified PCM value at the beginning of the path. This conditional probability density function is expressed as

$$p_{\underline{x}|x_1}(\alpha_2,\alpha_3,\alpha_4|\alpha_1 = A) = p_{\underline{x}}(A,\alpha_2,\alpha_3,\alpha_4)/p_{x_1}(A) \quad , \quad (4.2\text{-}8)$$

where the marginal density function is given as

$$p_{x_1}(\alpha_1) = \iiint_{-\infty}^{\infty} p_{\underline{x}}(\alpha_1,\alpha_2,\alpha_3,\alpha_4)\,d\alpha_2 d\alpha_3 d\alpha_4 \quad . \qquad (4.2\text{-}9)$$

Now we can calculate the probability of path r_1, r_2 and r_3. Let us call these path probabilities P_{r_1}, P_{r_2} and P_{r_3}, respectively. By tracing each path through the tree, with the aid of the step size algorithm for the Song audio mode ADM, we find that the probability of path r_1 can be expressed as

$$P_{r_1} = P(x(2)>S,\ x(3)>3S,\ x(4)<6S)$$
$$= \int_S^\infty \int_{3S}^\infty \int_{-\infty}^{6S} p_{\underline{x}|x_1}(\alpha_2,\alpha_3,\alpha_4|\alpha_1=A)\,d\alpha_2 d\alpha_3 d\alpha_4 \ . \tag{4.2-10}$$

Similarly, the probability of path r_2 is seen to be

$$P_{r_2} = P(x(2)>S,\ x(3)<3S,\ x(4)>2S)$$
$$= \int_S^\infty \int_{-\infty}^{3S} \int_{2S}^\infty p_{\underline{x}|x_1}(\alpha_2,\alpha_3,\alpha_4|\alpha_1=A)\,d\alpha_2 d\alpha_3 d\alpha_4 \ . \tag{4.2-11}$$

Finally, we formulate the probability of path r_3 as

$$P_{r_3} = P(x(2)<S,\ x(3)>S,\ x(4)>0)$$
$$= \int_{-\infty}^{S} \int_S^\infty \int_0^\infty p_{\underline{x}|x_1}(\alpha_2,\alpha_3,\alpha_4|\alpha_1=A)\,d\alpha_2 d\alpha_3 d\alpha_4 \ . \tag{4.2-12}$$

Now we apply the criterion that the most likely path has the highest path probability and state the path decision rule thusly:

$$\begin{aligned} &\text{if } P_{r_1} > P_{r_2} \text{ and } P_{r_3}, \quad \text{choose } r_1 , \\ &\text{if } P_{r_2} > P_{r_1} \text{ and } P_{r_3}, \quad \text{choose } r_2 , \\ &\text{if } P_{r_3} > P_{r_1} \text{ and } P_{r_2}, \quad \text{choose } r_3 . \end{aligned} \tag{4.2-13}$$

The chosen path is mapped into its $e_x(k)$ sequence and one cycle of the PCM to ADM conversion is completed. The process is subsequently repeated with the introduction of each new PCM sample.

This concept can be expanded to more than one Nyquist period, making the possible paths through the tree longer and taking advantage of the interdependence between the paths and several PCM samples. Alternately, we can modify the original procedure by not choosing an endpoint by applying Eq. (4.2-7). Instead, we might weight the path probabilities with a monotonically decreasing function of the endpoint error, ε, such as $1/\varepsilon$. Our most likely path decision will then be based on the largest weighted path probability.

There are, of course, many other variations on this theme. However, we stop here without an evaluation of the converter performance because we are searching for a <u>robust</u> system, not dependent on the probability statistics of the speaker. Using this system would require a "learning" period, during which time the machine determines the probability statistics of the speaker. This would have to be done for each speaker! Under these circumstances, this converter becomes impractical.

4.3 Parametric PCM-ADM Converter

A non-statistical method of solving the "multipath" problem is to first employ an estimation technique that introduces information about the signal between PCM samples. We can use this additional information and the PCM samples to generate a path through the estimate tree and produce the ADM bit stream. The parameter of the input signal that shall

be used to evaluate the converter performance and estimate the signal between PCM samples is the power spectral density of $x(t)$, $G_x(f)$. The estimation part of the converter can be designed entirely independent of the ADM employed. However, to generate the bit stream, $e_x(k)$, we must specify a step size algorithm for the ADM.

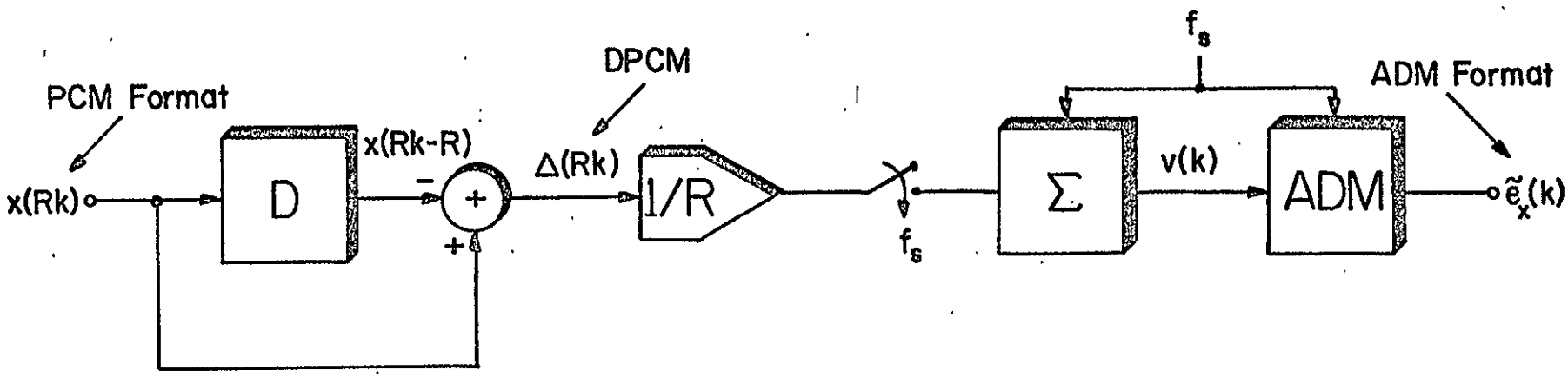
4.3.1 A Simple, All-Digital Technique

The most straightforward approach to achieve PCM to ADM conversion is to use an estimation technique which yields a linear approximation between PCM samples, take samples of this linear approximation signal at the ADM bit rate, and use these samples as the input to our Song audio mode ADM. We then automatically generate an ADM encoded signal, $e_x(k)$, and, at the same time, choose a path through the "DM signal estimate tree."

A completely digital circuit that can be easily implemented to perform this conversion is shown in Fig. 4.3-1. First, we form a differential PCM (DPCM) signal,

$$\Delta(Rk) = x(Rk) - x(R(k - 1)) \ , \qquad (4.3\text{-}1)$$

which is then digitally scaled by a factor $1/R$, sampled and accumulated at the ADM rate, f_s. The output of the accumulator, $v(k)$, is an equiamplitude staircase following the linear piecewise waveform formed by connecting the PCM samples. By ADM encoding $v(k)$, we complete the conversion. The scaling by $1/R$ can be introduced anywhere in the conver-



where $R = f_s/f_N$

$f_N = 1/T_N$ = Nyquist Rate

$D \longrightarrow T_N$ seconds delay

Fig. 4.3-1. A Basic PCM to ADM Converter

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

ter and can even be entirely eliminated without affecting the system performance. It is merely a constant multiplicative factor. We have included it in Fig. 4.3-1 merely for completeness so that v(k) can be accurately described. The sampling switch can be viewed as an AND gate being clocked at the ADM bit rate or, even more simply, as the accumulator clocking in the value $\Delta(Rk)$ R times every Nyquist period.

If the ADM were operated at a sufficiently high rate, the filtered DM estimate, denoted as w(t), would very closely approximate a linear piecewise function formed by connecting the PCM samples with straight lines. By viewing the estimation circuit as a digital filter which transforms the PCM samples into the staircase waveform, we can use the appropriate inverse filter after w(t) to obtain the PCM samples once again. Using this approach, the SNR of the conversion system approaches infinity.

This point of view is not very practical for a number of reasons. As mentioned before, the ADM bit rate is previously set by the value of R and normally we will encounter moderate values of f_s. The transformation from the PCM samples to w(t) must include the ADM, which performs a nonlinear operation. Thus, the true inverse transform is not readily realizable. As a more realistic approach, we shall use the piecewise linear waveform, w(t), without any inverse filtering, to obtain an upper bound on the SNR of the conversion system. This will be the basis of our analysis of the system performance and the calculation of in-band SNR.

4.3.2 Optimization of Converter Performance

The figure of merit of our converter will be the output, in-band signal-to-noise ratio, SNR_o. However, if we use the difference between x(t) and w(t) as the source of the output noise, this will yield somewhat misleading results. Without loss of generality, we can amplify and delay w(t) to improve SNR_o because constant attenuation and time delay cause no distortion in audio signals.

If we allow w(t) to be scaled by a factor, K, and delayed by a time, γ, then the error signal that we are concerned with becomes

$$d(t) = x(t) - Kw(t - \gamma) \ . \tag{4.3-2}$$

We now determine the autocorrelation function of the error signal as

$$R_d(\tau) \equiv E[d(t)d(t + \tau)] \ , \tag{4.3-3}$$

where

$$E(\alpha) = \text{the expectation of } \alpha.$$

By using the expression for d(t) given in Eq. (4.3-2), the autocorrelation function of the error can be expanded to

$$\begin{aligned} R_d(\tau) &= R_x(\tau) + K^2 R_w(\tau) \\ &\quad - K[R_{xw}(\tau - \gamma) + R_{xw}(\tau + \gamma)] \ , \end{aligned} \tag{4.3-4}$$

where

$R_x(\tau)$ = the autocorrelation function of x(t),

$R_w(\tau)$ = the autocorrelation function of w(t)

and

$R_{xw}(\tau)$ = the cross-correlation function of $x(t)$ and $w(t)$.

In deriving Eq. (4.3-4), we have made use of the fact, as will be shown later, that $w(t)$ is a linear function of the PCM samples of $x(t)$. A consequence of this is

$$E[w(t-\gamma)x(t+\tau)] \equiv R_{wx}(\tau+\gamma) = R_{xw}(\tau+\gamma) \quad , \quad (4.3\text{-}5)$$

or alternately, $R_{xw}(\alpha)$ is real and an even function of α.

To obtain the power spectral density of the error signal, we must take the Fourier transform of Eq. (4.3-4). The error signal power spectral density is thus seen to be

$$G_d(f) = G_x(f) + K^2 G_w(f) - 2K\cos(2\pi f\gamma)G_{xw}(f) \quad , \quad (4.3\text{-}6)$$

where

$G_x(f)$ = the Fourier transform of $R_x(\tau)$,

$G_w(f)$ = the Fourier transform of $R_w(\tau)$

and

$G_{xw}(f)$ = the Fourier transform of $R_{xw}(\tau)$.

Due to the property of $w(t)$ mentioned above, $G_{xw}(f)$ is also real and an even function of f. Finally, integrating over the bandwidth of $x(t)$, we find the in-band noise power, that is

$$P_d \equiv \int_{-f_m}^{f_m} G_d(f)\,df \quad . \quad (4.3\text{-}7)$$

Employing Eq. (4.3-6), the noise power is expressed as

$$P_d = P_x + K^2 P_w - 2K\int_{-f_m}^{f_m} \cos(2\pi f\gamma) G_{xw}(f)\,df \ , \qquad (4.3\text{-}8)$$

where P_x and P_w are defined in a manner similar to P_d in Eq. (4.3-7). The signal power, P_s, is defined as the in-band power contained in $Kw(t - \gamma)$, i.e.,

$$P_s = K^2 P_w \ . \qquad (4.3\text{-}9)$$

Using Eqs. (4.3-8) and (4.3-9), we can formulate the optimized figure of merit as

$$SNR_{opt} = Q \equiv P_s/P_d \ , \qquad (4.3\text{-}10)$$

which is found to be

$$Q = \frac{K^2 P_w}{P_x + K^2 P_w - 2K\mu(\gamma)} \ , \qquad (4.3\text{-}11)$$

where

$$\mu(\gamma) \equiv \int_{-f_m}^{f_m} \cos(2\pi f\gamma) G_{xw}(f)\,df \ . \qquad (4.3\text{-}12)$$

Now we seek to maximize Q by applying

$$\frac{\partial^2 Q(K,\gamma)}{\partial\gamma\,\partial K} = 0 \ . \qquad (4.3\text{-}13)$$

By setting $\partial Q/\partial\gamma$ equal to zero, we find that γ is a constant, denoted as γ_o, independent of K, which satisfies the relationship

$$\int_0^{f_m} f\sin(2\pi f\gamma_o) G_{xw}(f)\,df = 0 \ . \qquad (4.3\text{-}14)$$

The result of equating $\partial Q(K,\gamma_o)/\partial K$ to zero gives us the optimum K as

$$K_o = P_x/\mu(\gamma_o) \quad . \tag{4.3-15}$$

Finally, we have the maximum optimized SNR, $Q_o = Q(K_o,\gamma_o)$, expressable as

$$Q_o = \frac{P_x P_w}{P_x P_w - \mu^2(\gamma_o)} \quad , \tag{4.3-16}$$

where

$$\mu(\gamma_o) = \int_{-f_m}^{f_m} \cos(2\pi f \gamma_o) G_{xw}(f)\,df \quad . \tag{4.3-17}$$

We observe that the relationship given in Eq. (4.3-14) is satisfied when $\gamma_o = 0$. This corresponds to the expression for Q_o shown above. Since Q_o is to be a maximum value, the denominator should be minimized. This is achieved by picking the maximum of $\mu^2(\gamma_o)$, which occurs at $\gamma_o = 0$. To further support this last claim, we introduce a result which is derived later in this chapter. When w(t) is given as a linear function of the PCM samples of x(t), the cross-power spectral density, $G_{xw}(f)$, is proportional to the input power spectral density. Since $G_x(f)$ is bandlimited to f_m, so is $G_{xw}(f)$.

Because $G_{xw}(f)$ is real and even, we can use a slightly modified definition for the cross-correlation function [30],

$$R_{xw}(\tau) = \int_{-\infty}^{\infty} \cos(2\pi f \tau) G_{xw}(f)\,df \quad , \tag{4.3-18}$$

together with the fact that $G_{xw}(f)$ is bandlimited to f_m, to show that

$$\mu(\gamma_o) = R_{xw}(\gamma_o) \quad . \tag{4.3-19}$$

A simple property of this correlation function [31] is

$$|R_{xw}(\tau)| \leq R_{xw}(0) \quad . \tag{4.3-20}$$

Consequently, $\gamma_o = 0$ yields the maximum value of $\mu(\gamma_o)$ and the maximum value of Q_o. We can now rewrite Eq. (4.3-16) as

$$Q_o = \frac{P_x P_w}{P_x P_w - P_{xw}^2} \quad , \tag{4.3-21}$$

where

$$P_{xw} = \int_{-f_m}^{f_m} G_{xw}(f)\,df \quad . \tag{4.3-22}$$

Let us view the choice of γ_o from a practical viewpoint. Since we have defined w(t) as a linear piecewise signal formed by connecting the PCM samples of x(t), we really have not shifted w(t) with respect to x(t). If we recall, γ_o is the delay introduced in w(t) to optimize the SNR. But since w(t) has not been shifted at all, we naturally expect γ_o to be zero. In App. 4, we further pursue the dependence of Q_o on γ_o by working through an example where we obtain a plot of Q_o as a function of γ_o normalized by the Nyquist period, T_N.

4.3.3 SNR Statistical Analysis

To determine the SNR of our converter, we must first calculate the autocorrelation function of w(t) and the cross-correlation function of x(t) and w(t). To accomplish this we set up a coordinate system depicted in Fig. 4.3-2 and observe that the piecewise linear curve, w(t), is actually the sum of a sequence of ramps of slope m_i and the steps of amplitude w_i. To keep the time axis completely arbitrary, we introduce a random starting time for the ramp functions, shown as $-\tau_o$. Independent of this, we are insured that w(t) is a stationary random process, as long as x(t) is specified as being stationary, since it is obtained via a linear transformation on x(t). We introduce, in Fig. 4.3-2, a random variable, λ, which is uniformly distributed in the interval 0 to T_N. To completely define this coordinate system, we must specify the parameters:

$$i = \lceil \tau/T_N \rceil \equiv \text{the greatest integer} \leq \tau/T_N \,, \quad (4.3\text{-}23)$$

$$\tau = jT_N + \eta_o \,, \quad (4.3\text{-}24)$$

and

$$\eta = \tau - iT_N + \lambda = \eta_o + \lambda, \text{ for } 0 \leq \lambda \leq T_N \,. \quad (4.3\text{-}25)$$

Notice that η_o is just a positive constant in the interval $[0, T_N)$, which is specified by the choice of τ, and η is a random variable obtained via a linear transformation of λ.

We start by deriving the autocorrelation function of w(t). To facilitate matters, let us express the ramps at

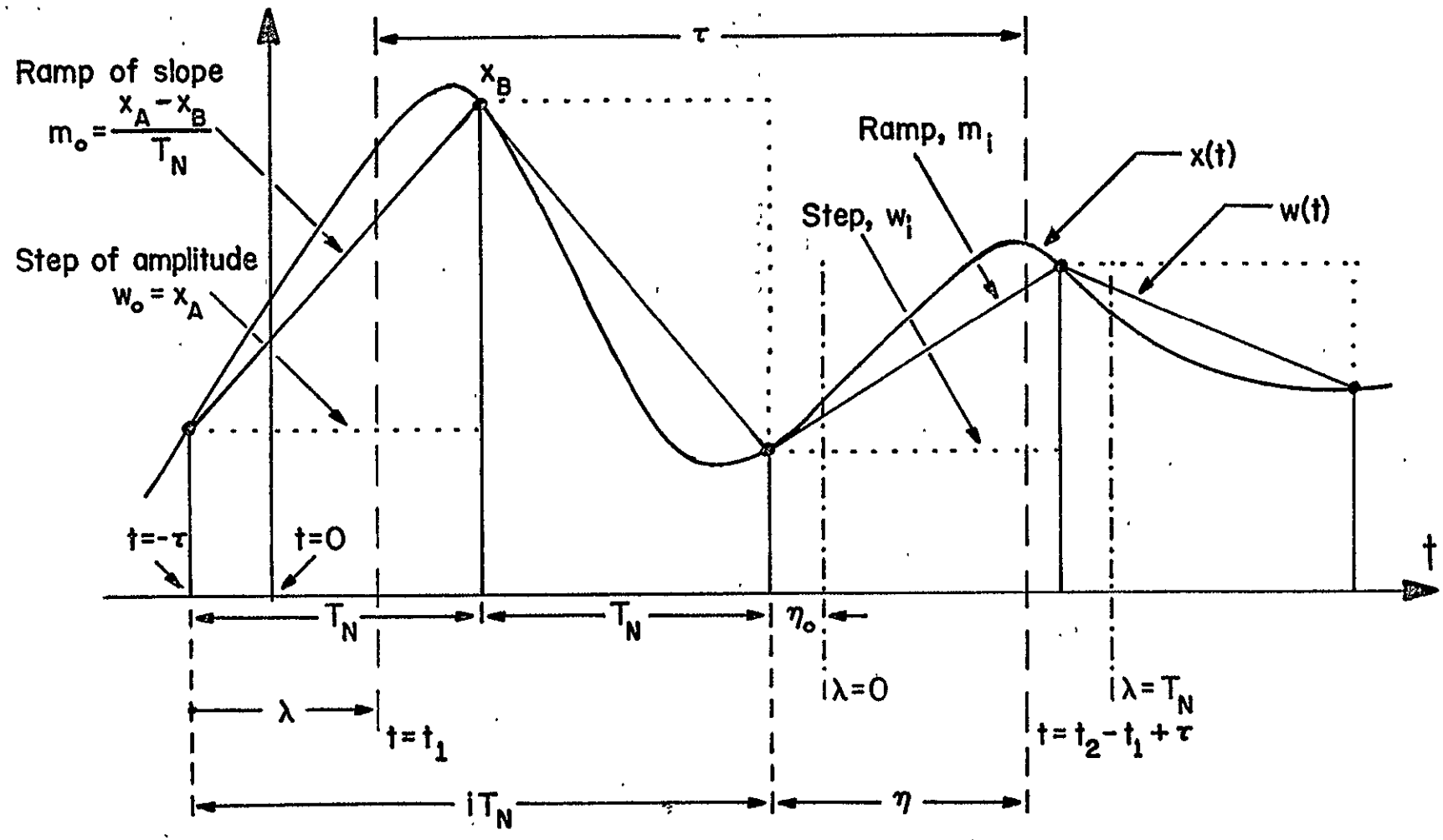


Fig. 4.3-2. Coordinate System for Converter SNR Analysis

times t_1 and t_2 as

$$z(t_1) = m_o \tag{4.3-26}$$

and

$$\begin{aligned} z(t_2 = t_1 + \tau) &= m_i \eta \qquad , \eta \leq T \\ &= m_{i+1}(\eta - T_N), \ \eta > T \ . \end{aligned} \tag{4.3-27}$$

The slope of the ramps can be expressed as

$$m_i = (w_{i+1} - w_i)/T_N \tag{4.3-28}$$

and the amplitude of the steps, w_i, is merely the sample values of $x(t)$, i.e.,

$$w_i = x_i \ . \tag{4.3-29}$$

By taking the expectation over both the ensemble of the random process $z(t)$ and the random variable λ, we can find the autocorrelation function of the ramps, that is

$$\begin{aligned} E[z(t_1)z(t_2)] = {} & (1/2T_N)\eta_o(T_N - \eta_o)^2 R_m(i) \\ & + (1/3T_N)(T_N - \eta_o)^3 R_m(i) \\ & + (1/2T_N)(\eta_o - T_N)[T_N^2 - (T_N - \eta_o)^2]R_m(i + 1) \\ & + (1/3T_N)[T_N^3 - (T_N - \eta_o)^3]R_m(i + 1) \ , \end{aligned} \tag{4.3-30}$$

where

$$\begin{aligned} R_m(i) &= E(m_k m_{k+i}) \\ &= [2R_x(i) - R_x(i + 1) - R_x(i - 1)]/T_N^2 \end{aligned} \tag{4.3-31}$$

and

$$R_x(i) = E(x_k x_{k+i}) \equiv R_x(iT_N) \ . \tag{4.3-32}$$

If we combine the ramps and the steps, then we can represent $w(t)$ at times t_1 and t_2 in the following way:

$$w(t_1) = w_o + m_o\lambda \tag{4.3-33}$$

and

$$\begin{aligned} w(t_2) &= w_i + m_i\eta \qquad\qquad , \ \eta \leq T_N \\ &= w_{i+1} + m_{i+1}(\eta - T_N), \ \eta > T_N \ . \end{aligned} \tag{4.3-34}$$

Using the averaging technique described above and the results found in Eq. (4.3-30), we can evaluate the autocorrelation function of $w(t)$, i.e.,

$$\begin{aligned} R_w(\tau) &= E[w(t_1)w(t_2)] \\ &= (1/T_N)(T_N - \eta_o)R_w(i) \\ &+ (1/6T_N)(T_N - \eta_o)^2(2T_N + \eta_o)R_m(i) \\ &+ (1/2T_N)(T_N - \eta_o)^2E(m_o w_i) \\ &+ (1/2T_N)(T_N^2 - \eta_o^2)E(m_i w_o) \\ &+ (\eta_o/T_N)R_w(i + 1) \\ &+ (\eta_o^2/6T_N)(3T_N - \eta_o)R_m(i + 1) \end{aligned}$$

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

$$+ (\eta_o/2T_N)(2T_N - \eta_o)E(m_o w_{i+1})$$

$$+ (\eta_o/2T_N)E(m_{i+1} w_o) \quad , \tag{4.3-35}$$

where

$$R_w(i) = E(w_k w_{k+i}) = R_x(i) \tag{4.3-36}$$

and

$$E(m_j w_k) = [R_x(k - j - 1) - R_x(k - j)]/T_N \ . \tag{4.3-37}$$

Simplifying Eq. (4.3-35), we arrive at a final form of the autocorrelation function of w(t),

$$R_w(\tau) = (\eta_o^3/6T_N^3)[3R_x(i) + R_x(i + 2)]$$

$$+ [(T_N - \eta_o)^3/6T_N^3][R_x(i - 1) + 3R_x(i + 1)]$$

$$- (1/T_N^2)[\eta_o^2 R_x(i) + (T_N - \eta_o)^2 R_x(i + 1)]$$

$$+ (2/3)[R_x(i) + R_x(i + 1)] \quad , \tag{4.3-38}$$

where

$$\eta_o = \tau - iT_N \tag{4.3-39}$$

and i is the greatest integer less than or equal to τ/T_N.

We can reformulate this final form of the autocorrelation function of w(t) such that it takes on a general structure and is a function of τ, that is,

$$R_w(\tau) = A_i\tau^3/T_N^3 + B_i\tau^2/T_N^2 + C_i\tau/T_N + D_i \ , \qquad (4.3\text{-}40)$$

where

A_i, B_i, C_i, D_i = real numbers which are a function of i and $R_x(i)$.

We see that $R_x(\tau)$ is a piecewise continuous function having different values of the coefficients A_i, B_i, C_i and D_i for each interval of τ, $[iT_N, (i + 1)T_N]$. A plot of $R_w(\tau)$, which shall be shown later, reveals that it is a well behaved autocorrelation function, being continuous, real and symmetric about $\tau = 0$.

Employing the notation for $w(t_1)$ introduced in Eq. (4.3-33), it becomes a less formidable task to calculate the cross-correlation function between $x(t)$ and $w(t)$. If we observe, from Fig. 4.3-2, that w_o and w_1 can be expressed as

$$w_o = x(t_1 - \lambda) \qquad (4.3\text{-}41)$$

and

$$w_1 = x(t_1 - \lambda + T_N) \ , \qquad (4.3\text{-}42)$$

then it can readily be shown that,

$$\begin{aligned} R_{xw}(\tau) = R_{wx}(\tau) &= E[w(t_1)x(t_1 + \tau)] \qquad (4.3\text{-}43) \\ &= \overline{(1 - \lambda/T_N)R_x(\tau + \lambda)} + \overline{(\lambda/T_N)R_x(\tau + \lambda - T_N)} \ , \end{aligned}$$

where the average in Eq. (4.3-43) is over the random variable λ. This can further be reduced to

$$R_{xw}(\tau) = (1/T_N)\int_0^{T_N} (1 - \lambda/T_N)[R_x(\tau + \lambda) + R_x(\tau - \lambda)]d\lambda \ . \tag{4.3-44}$$

Consequently, if the correlation function of x(t) is specified, we can formulate the correlation function of d(t), as given in Eq. (4.3-4), and its power spectral density, from Eq. (4.3-6). We can continue to simplify our results and show that the power spectral density obtained from $R_{xw}(\tau)$ is a function of the power spectral density of x(t). Taking the Fourier transform of Eq. (4.3-44) and performing some mathematical manipulations, we obtain, in its simplest form,

$$G_{xw}(f) = G_x(f)[\sin(\pi f T_N)/\pi f T_N]^2 \ . \tag{4.3-45}$$

To completely specify the power spectral density of d(t), we must calculate $G_w(f)$ from the correlation function of w(t). Since $R_w(\tau)$ takes the polynomial form given in Eq. (4.3-40), we can use a slightly modified version of a well known numerical technique for the evaluation of the Fourier transform to obtain an exact expression for $G_w(f)$ [32].

By taking the third derivative, with respect to τ, of the polynomial form of $R_w(\tau)$, given in Eq. (4.3-40), we obtain

$$R_w'''(\tau) = 6A_i/T_N^3 \ . \tag{4.3-46}$$

Recall that A_i takes on a different value for each interval of τ, $[iT_N, (i + 1)T_N]$. Thus, $R_w'''(\tau)$ takes on a different constant value every T_N seconds. Taking one more derivative

with respect to τ, we see that

$$T_N^3 R_W''''(\tau) = \sum_{i=-\infty}^{\infty} M_i \delta(\tau - iT_N) \ , \qquad (4.3\text{-}47)$$

where, owing to the symmetry of $R_W(\tau)$,

$$M_o = 12A_o \ , \qquad (4.3\text{-}48)$$

and

$$M_i = M_{-i} = 6(A_i - A_{i-1}), \text{ for } i = 1,2,3,\cdots \ , \qquad (4.3\text{-}49)$$

and

$\delta(\alpha) \equiv$ Dirac's delta function.

If we take the Fourier transform of Eq. (4.3-47), we obtain

$$\mathfrak{F}[T_N^3 R_W''''(\tau)] = T_N^3 (j\omega)^4 G_W(\omega) = \sum_{i=-\infty}^{\infty} M_i e^{-j\omega i T_N} \ . \qquad (4.3\text{-}50)$$

Reformulating this, we have

$$G_W(f) = [M_o + 2\sum_{i=1}^{\infty} M_i \cos(2\pi f i T_N)]/T_N^3 (2\pi f)^4 \ . \qquad (4.3\text{-}51)$$

Now that we have derived expressions for $G_{XW}(f)$ and $G_W(f)$, we must still evaluate P_{XW} and P_W to calculate the SNR. The unoptimized SNR is the expression for Q with K = 1 and $\gamma = 0$, that is, from Eq. (4.3-11),

$$SNR_o = \frac{P_W}{P_X + P_W - 2P_{XW}} \ . \qquad (4.3\text{-}52)$$

The optimized SNR, Q_o, is given in Eq. (4.3-21). Because

$G_{xw}(f)$ is bandlimited to f_m, we can easily evaluate P_{xw} via

$$P_{xw} = R_{xw}(0) = (2/T_N)\int_0^{T_N} (1 - \lambda/T_N)R_x(\lambda)d\lambda \ . \qquad (4.3\text{-}53)$$

However, the calculation of P_w is not so straightforward because $G_w(f)$ is not bandlimited and the value of $G_w(0)$ goes to infinity. Thus, we cannot integrate $G_w(f)$ from $-f_m$ to $+f_m$. Consequently, we must use

$$P_w = R_w(0) - P_{w_o} \ , \qquad (4.3\text{-}54)$$

where the out-of-band power in w(t) is given as

$$P_{w_o} \equiv 2\int_{f_m}^{\infty} G_m(f)df \qquad (4.3\text{-}55)$$

and the total power in w(t), $R_w(0)$, is found from Eq. (4.3-38) by setting η_o and i to zero. If we employ $G_w(f)$, as given in Eq. (4.3-51), then P_{w_o} takes the form

$$P_{w_o} = V_oM_o + \sum_{i=1}^{\infty} i^3V_iM_i \ , \qquad (4.3\text{-}56)$$

where V_i is a weighting coefficient that is a function of 0.5π - Si(i) and Si(α) is the sine integral, defined in Eq. (3.3-15).

We shall demonstrate the results of this statistical analysis by evaluating the SNR for two examples. In the first example, the input signal has a white, bandlimited

power spectral density and in the second example the input power spectral density has a triangular shape. For the former case, the sum in Eq. (4.3-56) is finite. For the latter case, we can truncate after only 5 terms because the series converges rapidly to zero. In the next part, where we introduce improvements on this basic PCM to ADM conversion system, the series does not converge so rapidly and we must introduce an estimation procedure to evaluate P_{w_o} to avoid excessive laborious calculation.

If we specify unity input power, the power spectral density of the white, bandlimited process (E.g. 1) is given as

$$G_{1x}(f) = T_N, \ |f| < f_m$$
$$= 0 \ , \ |f| > f_m \ , \qquad (4.3\text{-}57)$$

and its correlation function is

$$R_{1x}(\tau) = \sin(2\pi f_m \tau)/2\pi f_m \tau \ . \qquad (4.3\text{-}58)$$

For this example, the output SNR and the optimized SNR is calculated to be

$$SNR_o(\text{E.g. } 1) = 8.6 \text{ dB}$$

and

$$Q_o(\text{E.g. } 1) = 12.7 \text{ dB} \ .$$

When the input spectrum has a triangular shape and we again set the input power to unity (E.g. 2), the power spectral density takes the form

$$G_{2x}(f) = 2T_N(1 - |f/f_m|), \quad |f| \leq f_m$$
$$= 0 \qquad\qquad , |f| \geq f_m \ . \tag{4.3-59}$$

For this process, the autocorrelation function is seen to be

$$R_{2x}(\tau) = [\sin(\pi f_m \tau)/\pi f_m \tau]^2 \ . \tag{4.3-60}$$

The figures of merit for this example are evaluated as

$$SNR_o(\text{E.g. } 2) = 14.2 \text{ dB}$$

and

$$Q_o(\text{E.g. } 2) = 17.3 \text{ dB} \ .$$

Obviously, this level SNR is not acceptable, even for the extreme case of a white, bandlimited input power spectrum. Therefore, we must introduce a refinement of this basic conversion technique to improve the performance and yield larger values of SNR.

4.3.4 Converter Improvement via Wiener Linear Interpolation

The performance of this converter can be improved with a parametric technique, by first estimating a PCM midpoint from adjacent PCM samples, that is, both past and future values, and then converting to ADM format. The structure of the improved converter is basically the same as before and is shown in Fig. 4.3-3. We form the difference between the midpoint estimate and the PCM sample at the beginning of the Nyquist interval, scale this value by 2/R and accumulate it

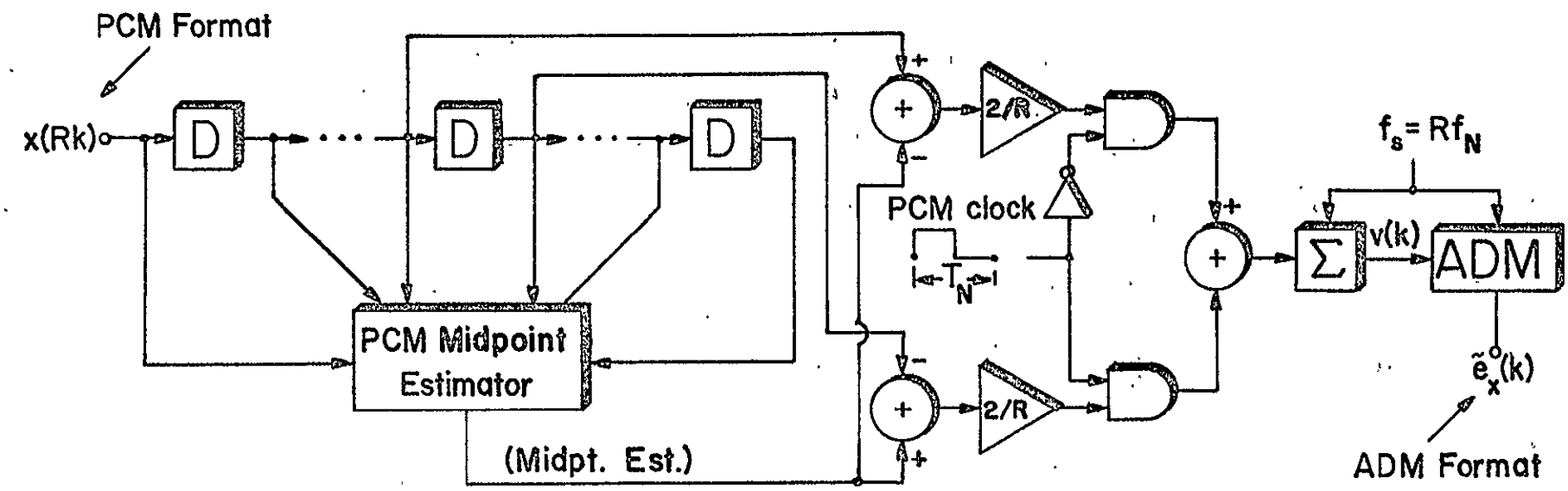


Fig. 4.3-3. Improved PCM to ADM Converter

R/2 times during the first half of the PCM clock period. This forms the signal v(k) which is encoded by the ADM. Then we formulate the difference between the PCM value at the end of the Nyquist interval and the midpoint estimate. This value is scaled by 2/R and accumulated R/2 times during the last half of the PCM clock period. This continues to produce the v(k) signal which is again ADM encoded into the sequence of bits, $e_x(k)$. This process will yield two equi-amplitude staircases of different slopes for v(k) during each PCM sampling period.

A Wiener linear interpolation technique is employed to obtain the PCM midpoint estimate since it minimizes the mean-square error [33]. The midpoint estimate is formulated as a linear combination of the adjacent PCM samples. The weighting coefficients in the linear combination are a function of the autocorrelation function of the input process. The improvement is demonstrated by analyzing this converter when 2 and then 4 adjacent PCM samples are used to estimate the midpoint. The resulting SNR is evaluated for the two examples mentioned above, that is, two different input power spectral densities. To determine a bound on the performance of this converter, we calculate the SNR limit by letting the estimated PCM midpoint actually become the true PCM sample.

The basic theory of linear mean-square estimation is best presented using a general approach. Since we have assumed that x(t) is a stationary random process, then all samples of it will be random variables. We must estimate

the midpoint between two samples from these random variables. If we use 2I sample points, then the midpoint estimate can be formulated as a linear combination of these random variables in the following way:

$$\begin{aligned}\tilde{x}_p(Rk + R/2) &= a_{-(I-1)}x(Rk - (I-1)R) + \ldots \\ &+ a_{-1}x(Rk - R) + a_0 x(Rk) \\ &+ a_1 x(Rk + R) + \ldots + a_I x(Rk + IR) \ .\end{aligned} \tag{4.3-61}$$

Using a shorthand notation, we can express the midpoint estimate as

$$\tilde{x}_p = \sum_{i=-(I-1)}^{I} a_i x_i \ . \tag{4.3-62}$$

To find the values of a_i we apply the orthogonality principle. This principle states that the error must be orthogonal to the random variables. In other words,

$$\begin{aligned}&E[(x_p - \tilde{x}_p)x_j] = 0 \ , \\ &\text{for } j = -(I-1), \ldots, -1, 0, 1, \ldots, I \ .\end{aligned} \tag{4.3-63}$$

After taking the expected value, we obtain

$$\begin{aligned}&R_x(j - 0.5) = \sum_{i=-(I-1)}^{I} a_i R_x(j - i) \ , \\ &\text{for } j = -(I-1), \ldots, -1, 0, 1, \ldots, I \ ,\end{aligned} \tag{4.3-64}$$

which is merely a set of 2I equations with 2I unknowns. Thus, we see that the values of a_i depend on the autocorre-

lation function of the input process.

Now we can investigate the PCM to ADM converter performance when the midpoint is estimated with 2 and 4 adjacent PCM samples, this is, when I = 1 and 2, respectively. The statistical analysis deals with w(t), which is a straight line approximation between PCM points and midpoint estimates. To facilitate the SNR analysis, we divide the time axis into intervals of width $T_N/2$ and index them as follows:

$$n = 0, 1, 2, \ldots \qquad \text{for } t \geq 0 ,$$
$$= -1, -2, -3, \ldots \qquad \text{for } t < 0 .$$

We can again represent w(t) as the sum of steps and ramps. However, the form of the step and the ramp will be different for n odd and even. But, they will always be a function of the adjacent PCM samples due to the linear interpolation technique used to formulate the midpoint estimates.

To determine $R_w(\tau)$, we can formulate $w(t_1)$ and $w(t_2)$ in a manner similar to Eqs. (4.3-33) and (4.3-34), except $w(t_1)$ now spans two intervals and $w(t_2)$ spans three because the width of each interval is only $T_N/2$. Now we must calculate $R_w(\tau)$ for n odd and n even, since the value of τ sets n according to

$$n = \lceil 2\tau/T_N \rceil = \text{the greatest integer} \leq 2\tau/T_N. \qquad (4.3\text{-}65)$$

Also, the amount of computation that must be performed increases by several orders of magnitude and cannot be readily computerized because there is extensive algebraic manipulations.

When the midpoint is estimated by 2 adjacent PCM samples, the autocorrelation function takes the form

$$R_w(\tau) = C_{-2}R_x(n-2) + C_0R_x(n) + C_2R_x(n-2) + C_4R_x(n+4) \tag{4.3-66a}$$

for n even, and

$$R_w(\tau) = C_{-3}R_x(n-3) + C_{-1}R_x(n-1) + C_1R_x(n+1) + C_3R_x(n+3) \tag{4.3-66b}$$

for n odd, where

C_j = a third order polynomial function of η_o and a function of a_i.

For this converter, we use

$$\eta_o = \tau - nT_N/2 \ . \tag{4.3-67}$$

From the form of $R_w(\tau)$ given above, it can be expressed as a third order polynomial function of τ and, therefore, we can apply the technique discussed in Sec. 4.3.3 to calculate the power spectral density of w(t) and the in-band power, P_w.

The evaluation of the cross-correlation function is approached exactly as described in the previous section. Because $w(t_1)$ spans two intervals, the amount of computation greatly increases in determining $R_{xw}(\tau)$. After several pages of calculations, the final form becomes

$$R_{xw}(\tau) = (2/T_N)\int_0^{T_{N/2}} (2 - \alpha_1\lambda/T_N)$$

$$\cdot\ (R_x(\tau + \lambda) + R_x(\tau - \lambda))d\lambda$$

$$+ (2/T_N)\int_{T_{N/2}} \alpha_2(1 - \lambda/T_N) \quad . \qquad (4.3\text{-}68)$$

$$\cdot\ (R_x(\tau + \lambda) + R_x(\tau - \lambda))d\lambda \ ,$$

where

$$\alpha_1,\ \alpha_2 = \text{a function of } a_i .$$

Taking the Fourier transform of Eq. (4.3-68), we again find that $G_{xw}(f)$ is bandlimited to f_m so that we can calculate the in-band cross-power via

$$P_{xw} = R_{xw}(0) \ . \qquad (4.3\text{-}69)$$

The figures of merit can now be evaluated, using Eq. (4.3-25) to determine SNR_o and Eq. (4.3-21) for Q_o. For example 1, where the power spectral density is white and bandlimited, we obtain

$$SNR_o(\text{E.g. } 1) = 10.2 \text{ dB}$$

and

$$Q_o(\text{E.g. } 1) = 11.7 \text{ dB} \ .$$

When the input spectrum is triangular shaped, we find that

$$SNR_o(\text{E.g. } 2) = 15.5 \text{ dB}$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

and

$$Q_o(\text{E.g. } 2) = 16.2 \text{ dB} .$$

The slight increase in performance over the basic converter without a midpoint estimate indicates that our interpolation of the midpoint is not nearly accurate enough. We shall consider one more case, that is, estimating the midpoint from 4 adjacent PCM samples. Then we shall analyze the limiting case, where the estimated midpoint actually becomes the true PCM sample.

The statistical analysis, when the midpoint is estimated from 4 adjacent PCM points, takes the exact structure as described previously. Thus, all we shall do is present an outline of the results. The autocorrelation function of $w(t)$ takes the form:

$$R_w(\tau) = \sum_{j=-3}^{4} C_{2j} R_x(n + 2j), \text{ for n even,} \qquad (4.3\text{-}70a)$$

and

$$R_w(\tau) = \sum_{j=-3}^{4} C_{2j-1} R_x(n + 2j - 1), \text{ for n odd,} \qquad (4.3\text{-}70b)$$

where C_j is given after Eqs. (4.3-66).

The effect of employing more PCM samples to estimate the midpoint manifests itself in the dependence of $R_w(\tau)$ on the input autocorrelation function. The cross-correlation function is structured exactly as in Eq. (4.3-68) except that now λ is integrated over 4 intervals, of width $T_N/4$,

in going from $\lambda = 0$ to $\lambda = T_N$. Consequently, $G_{xw}(f)$ is again bandlimited to f_m and we can use Eq. (4.3-69) to determine P_{xw}. As before, the in-band power, P_w, is found by applying the technique discussed in Sec. 4.3.3. Through all this analysis, the important results are still the figures of merit. For the white, bandlimited power spectral density we obtain

$$SNR_o(\text{E.g. } 1) = 13.0 \text{ dB}$$

and

$$Q_o(\text{E.g. } 1) = 14.3 \text{ dB} .$$

If the input spectrum is triangular shaped, the results are

$$SNR_o(\text{E.g. } 2) = 17.8 \text{ dB}$$

and

$$Q_o(\text{E.g. } 2) = 18.2 \text{ dB} .$$

Although the performance of this converter will continue to improve as more PCM samples are used to estimate the midpoint, we would like to know the maximum value of the SNR. To determine the maximum SNR, we must consider the limiting case where the estimated midpoint actually becomes the true PCM sample. The statistical analysis for the limiting case has already been completed. When the estimated midpoint becomes the true PCM sample the entire structure of the converter reverts back to the basic PCM to ADM converter except the PCM rate is now $2f_N$. Therefore, the results are identical to those derived in Sec. 4.3.3, but we must replace

T_N by $T_N/2$. This must be done <u>everywhere</u>, so the meaning of $R_x(n)$ now becomes

$$R_x(n) = R_x(\tau = nT_N/2) \quad . \tag{4.3-71}$$

Returning to the expressions for $G_w(f)$ and $R_{xw}(\tau)$ and changing T_N to $T_N/2$, it becomes simply an arithmetic task to calculate P_w and P_{xw}. The results for the white, bandlimited and the triangular shaped power spectral densities are as follows:

$$SNR_o(\text{E.g. } 1) = 21.1 \text{ dB}$$

$$Q_o(\text{E.g. } 1) = 25.3 \text{ dB}$$

and

$$SNR_o(\text{E.g. } 2) = 26.4 \text{ dB}$$

$$Q_o(\text{E.g. } 2) = 29.3 \text{ dB} \quad .$$

Now the SNR is beginning to reach moderate levels. However, it is only at the expense of very precise estimation of the PCM midpoint. In the next section, we consider further improvement of this PCM to ADM converter and a simple estimation technique to calculate SNR.

4.3.5 Performance of the Improved Converters

A further refinement of this conversion technique is obtained by estimating two PCM values between adjacent PCM samples and employing all of these points to convert to ADM format. The two estimated PCM values are obtained, as before, using the Wiener criterion. In this case, we have

three equiamplitude staircases during each PCM sampling interval. The SNR analysis can easily be extended to this system just as was done when only the midpoint was estimated. However, the algebraic computation becomes much too extensive. To ascertain the performance that we achieve with this improved method, we can evaluate the limiting SNR for the two input power spectral densities cited above.

If we wish to apply the original statistical analysis to the limiting case for 2, 3 or m estimated points between adjacent PCM samples, all we must replace T_N by T_N/m, in all the results derived in Sec. 4.3.3. However, in calculating the output signal power, P_W, we begin to experience computational difficulties. Since P_W was formulated as the difference between the total power and the out-of-band power, P_{W_O}, we must calculate P_{W_O} from an infinite sum of terms via Eq. (4.3-56). When we estimate more PCM points and take the limiting case, P_{W_O} requires more and more terms to yield meaningful results. These terms are a function of the sine integral. The result now depends on the accuracy of the sine integral tables or computation.

To circumvent this computational problem, we introduce a simple estimation technique to calculate SNR. We start with the expression for the optimized SNR, where the input power has been normalized to unity, i.e.,

$$Q_O = \frac{P_W}{P_W - P_{XW}^2} . \qquad (4.3\text{-}72)$$

We note that all terms in the expression for Q_o are just slightly less than unity, particularly for the limiting cases where we estimate 2 or more points between PCM samples. Consequently, the denominator of Eq. (4.3-72) becomes the dominant term. Since Q_o can never be negative, the smallest value of P_w is

$$P_w(\min) = P_{xw}^2 \ . \tag{4.3-73}$$

Therefore, the upper limit on the out-of-band power of w(t) becomes

$$P_{w_o}(\max) = R_v(0) - P_{xv}^2 \ . \tag{4.3-74}$$

Since the lower limit on P_{w_o} is zero, we can express it as

$$P_{w_o} = \alpha P_{w_o}(\max), \qquad 0 < \alpha < 1 \ . \tag{4.3-75}$$

The objective of this estimation technique is to eliminate P_w so there is no need to calculate P_{w_o}. We can then arrive at an expression for the SNR which is composed of terms that are easily evaluated. Using Eqs. (4.3-74) and (4.3-75) we can formulate the output in-band power as

$$P_w = (1 - \alpha)R_w(0) + \alpha P_{xw}^2 \ . \tag{4.3-76}$$

We can now obtain the desired figures of merit without determining P_w. Setting the input power to unity, we obtain the following expressions for the SNR, before and after optimization,

$$SNR_o = \frac{(1 - \alpha)R_w(0) + \alpha P_{xw}^2}{1 + (1 - \alpha)R_w(0) + \alpha P_{xw}^2 - 2P_{xw}} \qquad (4.3\text{-}77)$$

and

$$Q_o = \frac{(1 - \alpha)R_w(0) + \alpha P_{xw}^2}{(1 - \alpha)(R_w(0) - P_{xw}^2)} \qquad (4.3\text{-}78)$$

Fitting these expressions to the results obtained so far, we have found that a value of $\alpha = 0.5$ yields a very good approximation.

The performance of all PCM to ADM converters discussed so far, for the two input power spectral densities cited above, is summarized in Table 4.3-1. Here we list SNR_o and Q_o through the limiting case when three points are estimated between adjacent PCM samples. Applying the SNR estimation technique described above, we have extended the results to seven estimated PCM points. In Fig. 4.3-4, we plot SNR_o in dB versus the number of estimated PCM points, for the limiting case, for both E.g. 1 and E.g. 2. As expected, each further refinement yields a higher SNR than all other converters previously considered. The tradeoff is, of course, an increase in hardware complexity.

To see the progressive change in converter performance, with each subsequent modification, we can examine the autocorrelation function of w(t). In Figs. 4.3-5 and 4.3-6, we have plotted $R_w(\tau)$ for E.g. 1 and E.g. 2, respectively, for three different PCM to ADM converter structures. As we go

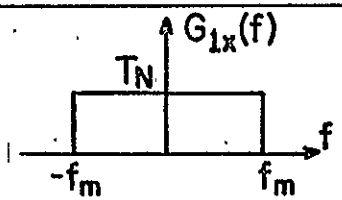
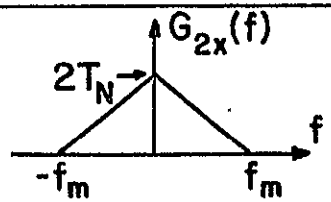
STRAIGHT LINE APPROXIMATION BETWEEN PCM AND ESTIMATED PCM POINTS					
NO. OF ESTIMATE POINTS	NO. OF ADJACENT PCM POINTS	E.g. 1 		E.g.2 	
		SNR_o (dB)	Q_o (dB)	SNR_o (dB)	Q_o (dB)
0	-	8.6	12.7	14.2	17.3
1	2	10.2	11.7	15.5	16.2
1	4	13.0	14.8	17.8	18.2
1	Limiting Case	21.1	25.3	26.4	29.3
2	Limiting Case	28.3	32.6	33.5	36.6
3	Limiting Case	33.3	37.6	38.5	41.6

TABLE 4.3-1. Performance of Parametric PCM to ADM Converters

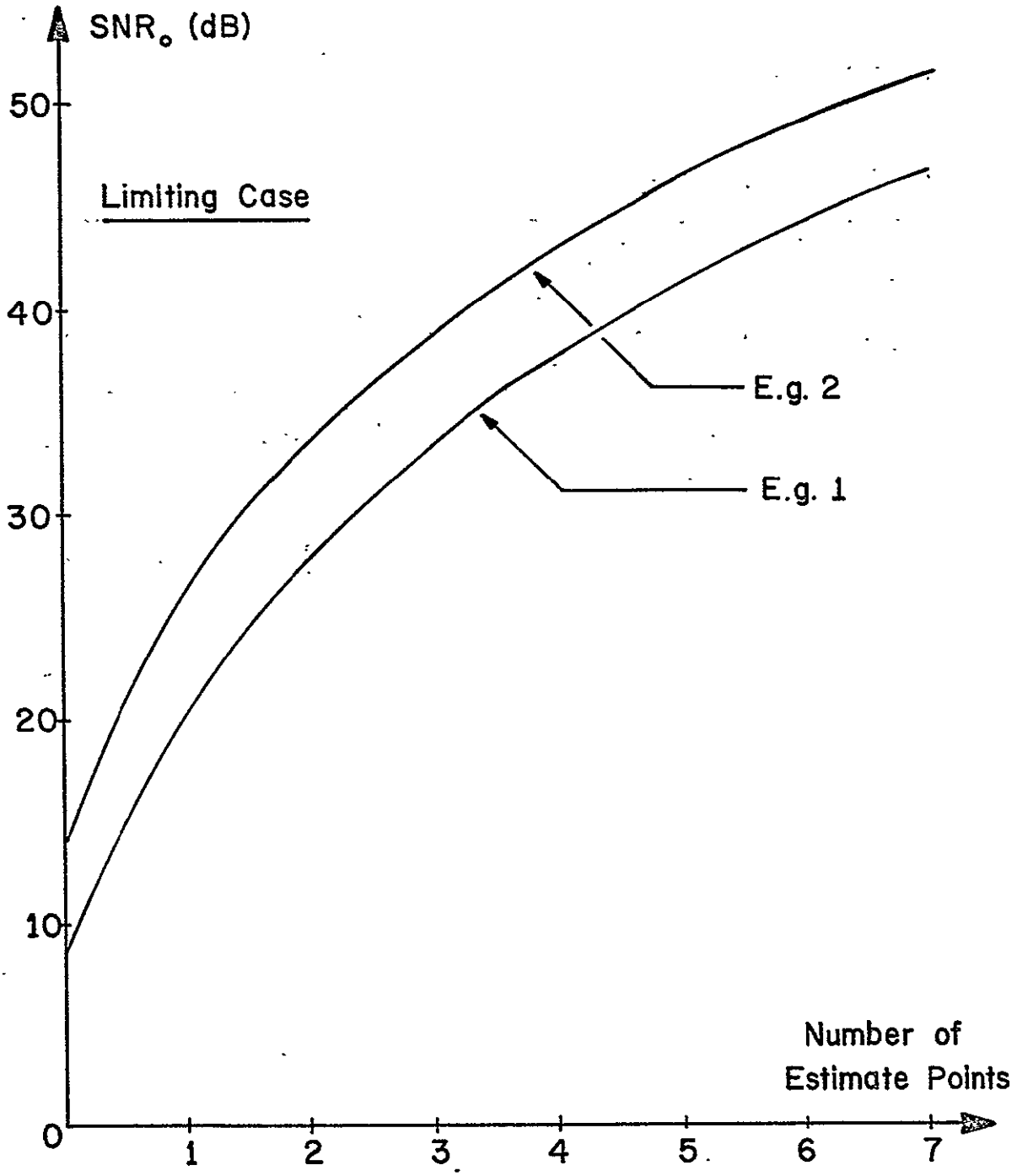


Fig. 4.3-4. SNR of Improved PCM to ADM Converters for the Limiting Case

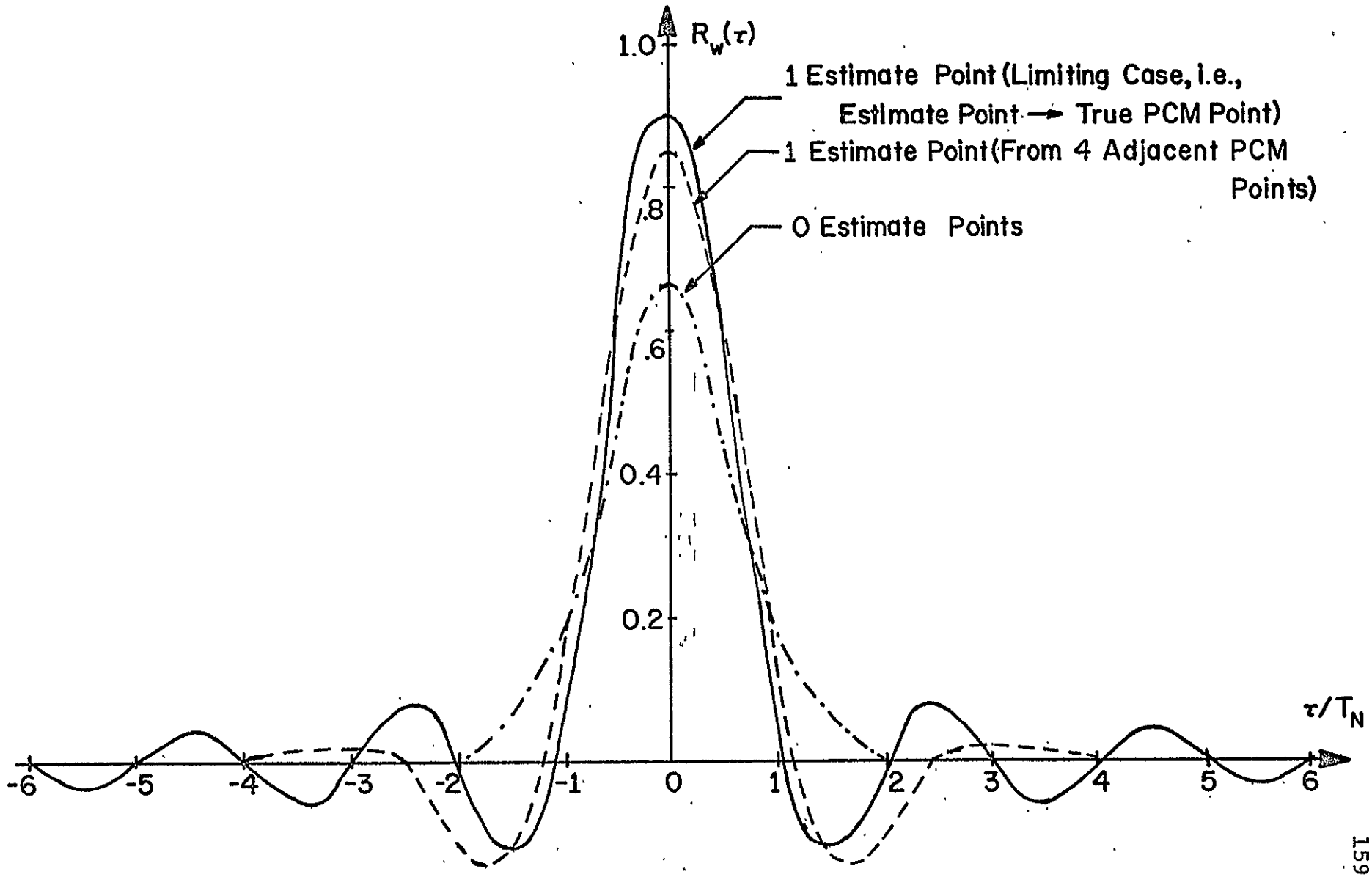


Fig. 4.3-5. Correlation Function of w(t) when Input Power Spectrum is White and Bandlimited

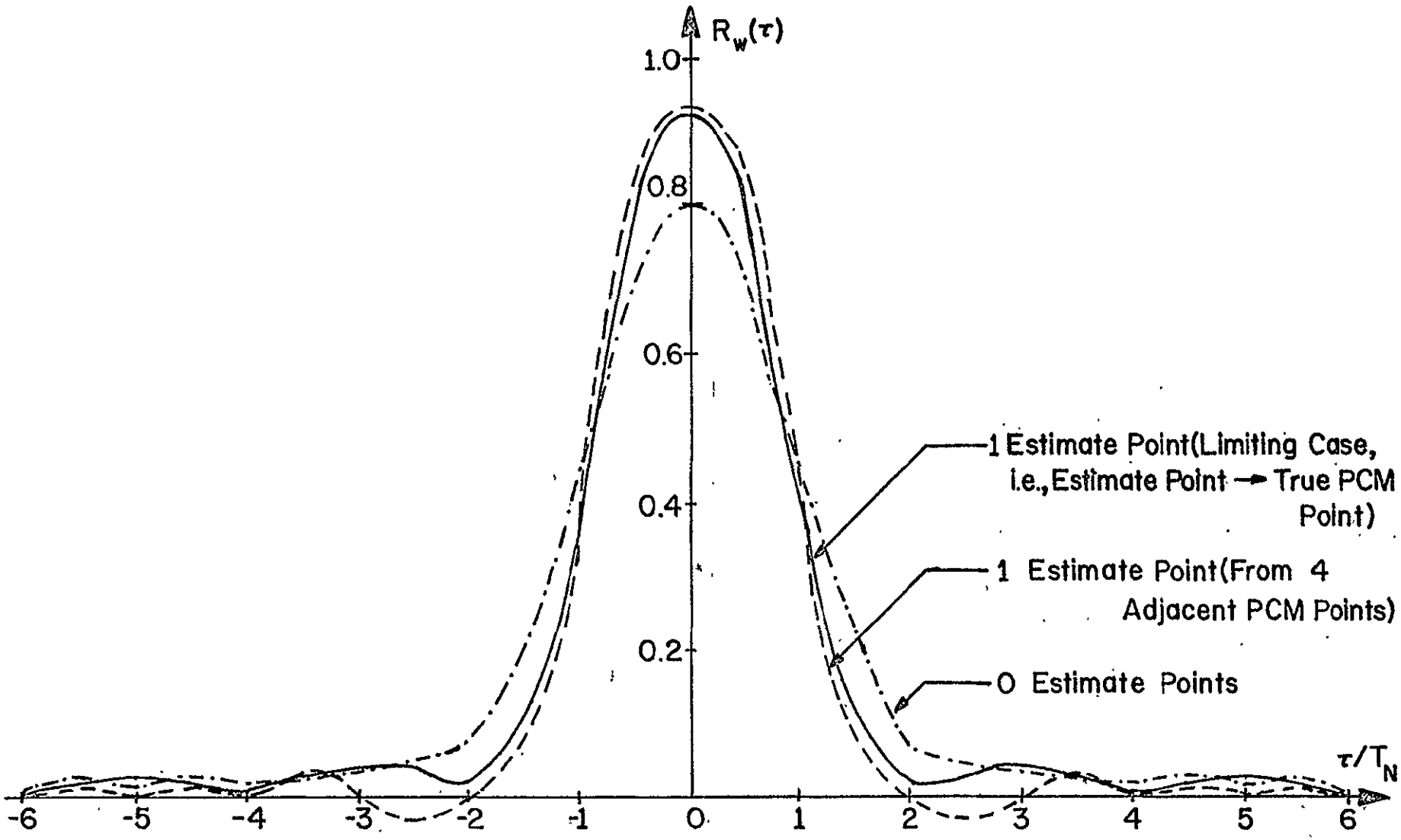


Fig. 4.3-6. Correlation Function of w(t) when Input Power Spectrum is Triangular

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

from the basic converter where no PCM points are estimated, to the system which estimates a midpoint from four adjacent PCM samples, to the limiting case for one estimate point, we observe that $R_w(\tau)$ yields a better and better approximation for the input autocorrelation functions, defined in Eqs. (4.3-58) and (4.3-60), for these two examples. The closer the autocorrelation function of w(t) is to $R_x(\tau)$, the nearer P_w and P_{xw} will be to P_x. This, of course, gives us a higher and higher output SNR. Therefore, the shape of $R_w(\tau)$ acts as a very good indication of the performance of our PCM to ADM converter.

4.3.6 Simulation of the Fundamental Converter

The basic PCM to ADM converter, as shown in Fig. 4.3-1, was simulated using a PDP 8/L computer. The Song audio mode ADM algorithm was employed in this simulation and several different elementary input signals were used to test its operation. To evaluate the performance of this converter, we utilized an ADM decoder, which is merely the feedback circuit of the encoder, and simulated a fourth-order Butterworth filter as the output LPF. A sinusoidal test tone was applied to the input and SNR was measured after the output LPF. The simulation system is given in Fig. 4.3-7. Note that the cutoff frequency of the LPF, f_c, sets the bandwidth of the system and, therefore, the PCM sampling rate used at the input becomes

$$f_N = 2f_c \ . \tag{4.3-79}$$

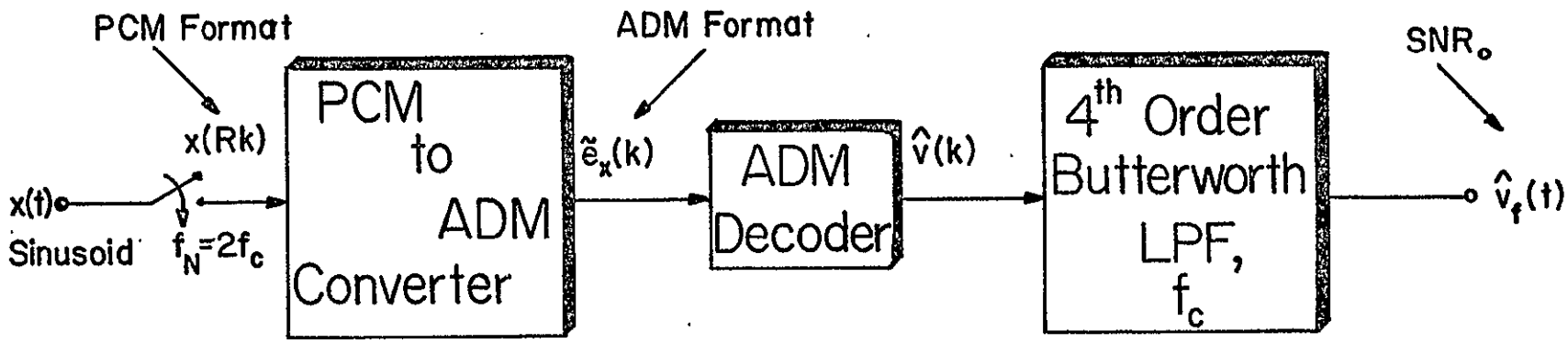


Fig. 4.3-7. Simulation System for PCM to ADM Converter Performance

In all previous performance evaluations, we employed a test tone of frequency f_o and specified the LPF cutoff frequency to be $4f_o$. This, however, is not sufficient to accurately test the performance of this converter. In Fig. 4.3-8, we show a low and a high test tone, at frequencies f_o and $3f_o$, respectively. Both inputs must be utilized to ascertain a true picture of the operation of this system. On the low and high tones, we have superimposed the PCM samples and the straight line approximations between them which represent w(t). To gain a further idea of the system performance with the low tone, we have calculated the SNR of v(k) for various values of R. Recall that v(k) is the staircase waveform between PCM points and R, defined in Eq. (4.2-1), is actually the number of steps between PCM points.

We display, in Fig. 4.3-9, SNR_v, which is the SNR for v(k), after final low pass filtering, as a function of R. Since w(t) follows x(t) so well, and because SNR_v is relatively high, we do not expect much degradation when ADM converting v(k). We expect this test tone to yield good performance curves, indicating that the PCM to ADM converter is a success without using any PCM estimate points between adjacent samples. However, for the high tone, we observe that w(t) is a poor approximation to x(t). It is so poor, in fact, that without even calculating SNR_v, we know that we must use, at least, an estimated midpoint to achieve acceptable performance. But, we cannot estimate a midpoint based on the theory presented previously because it employed probabalistic

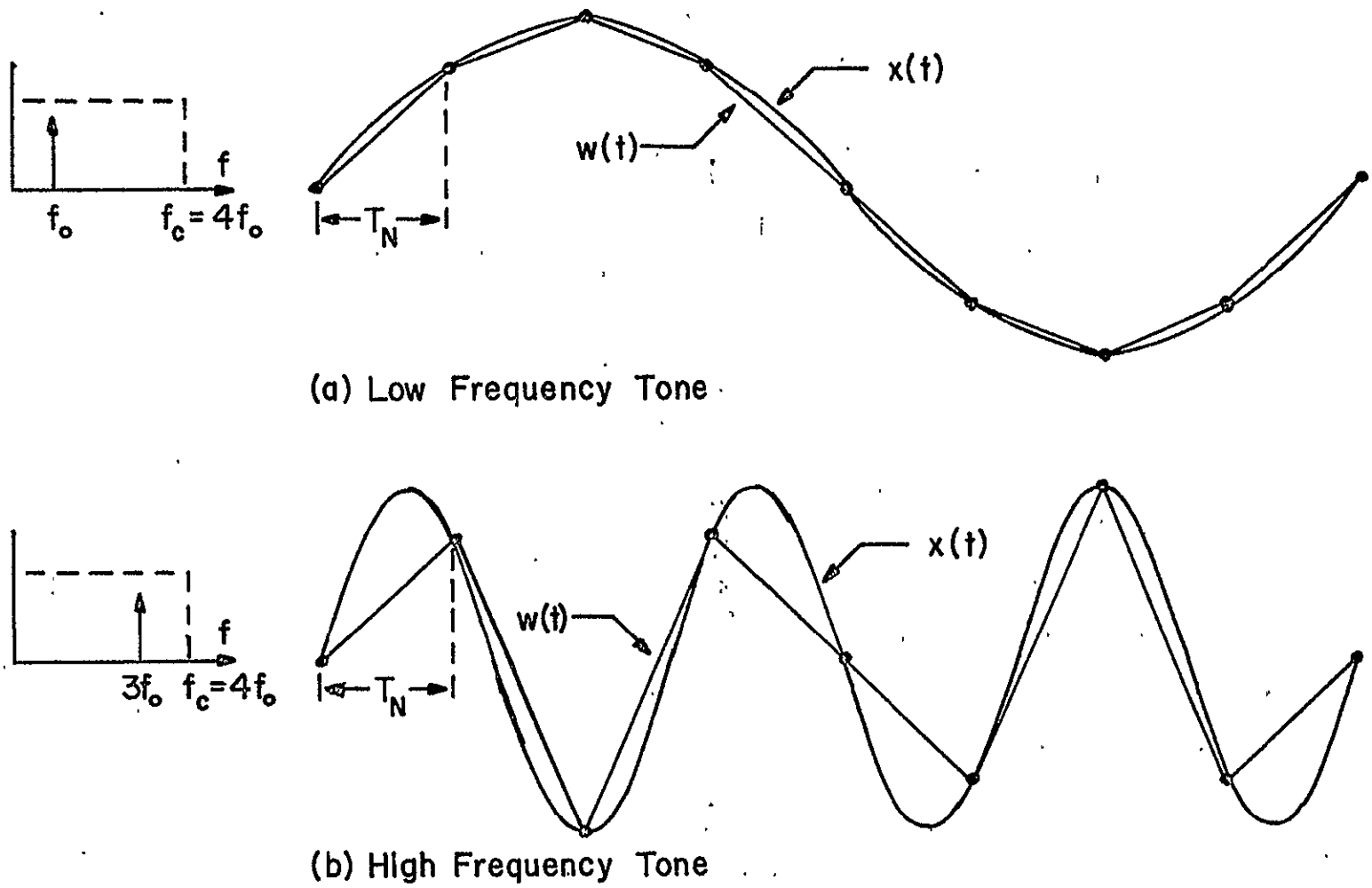


Fig. 4.3-8. Test Tones for PCM to ADM Converter and Straight Line Approximations between PCM Points

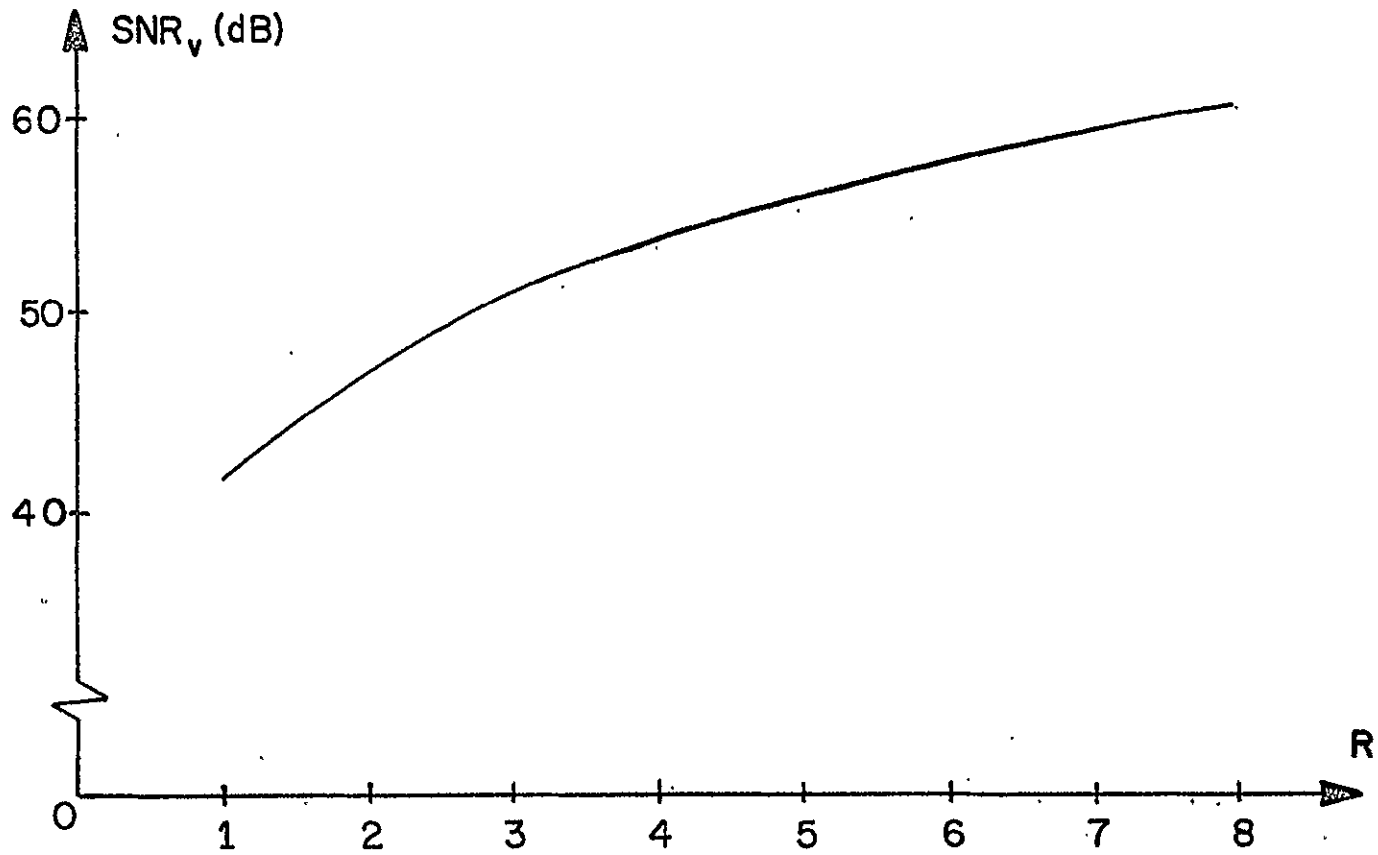


Fig. 4.3-9. SNR of Staircase Waveform as a Function of the Number of Steps between PCM Points

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

parameters of the input signal. These are virtually meaningless for a deterministic signal like a sinusoidal input. In the next section, we shall present the quantitative results obtained from these simulations and then go on to introduce a non-parametric technique to estimate points which can easily be applied to the sinusoidal inputs.

4.3.7 SNR Curves

All techniques, assumptions and manipulations used in Chs. 2 and 3 to determine output SNR, when a sinusoidal input is applied to a computer simulated system, will again be utilized with the basic PCM to ADM converter. Since the input PCM samples, for both the low and high tones, are periodic, $v(k)$ is also periodic and so is the resulting ADM estimate, $\hat{v}(k)$. We must first determine the Fourier series of the estimate. Then we can LPF $\hat{v}(k)$ by scaling the frequency components as described in Sec. 2.8. Finally, we can calculate the output performance figure of merit, SNR_o, from Eq. (2.8-12), using the output noise power truncated after the ninth harmonic.

By shifting the input $x(t)$ with respect to the sampling clock, we again generated different steady state estimate patterns for $\hat{v}(k)$ and, therefore, different values of SNR_o. Treating the output SNR as a random variable, we calculated its mean and variance for 32 different starting points of the input sinusoid. In Fig. 4.3-10, we display the performance curves for the basic PCM to ADM converter, i.e., with no

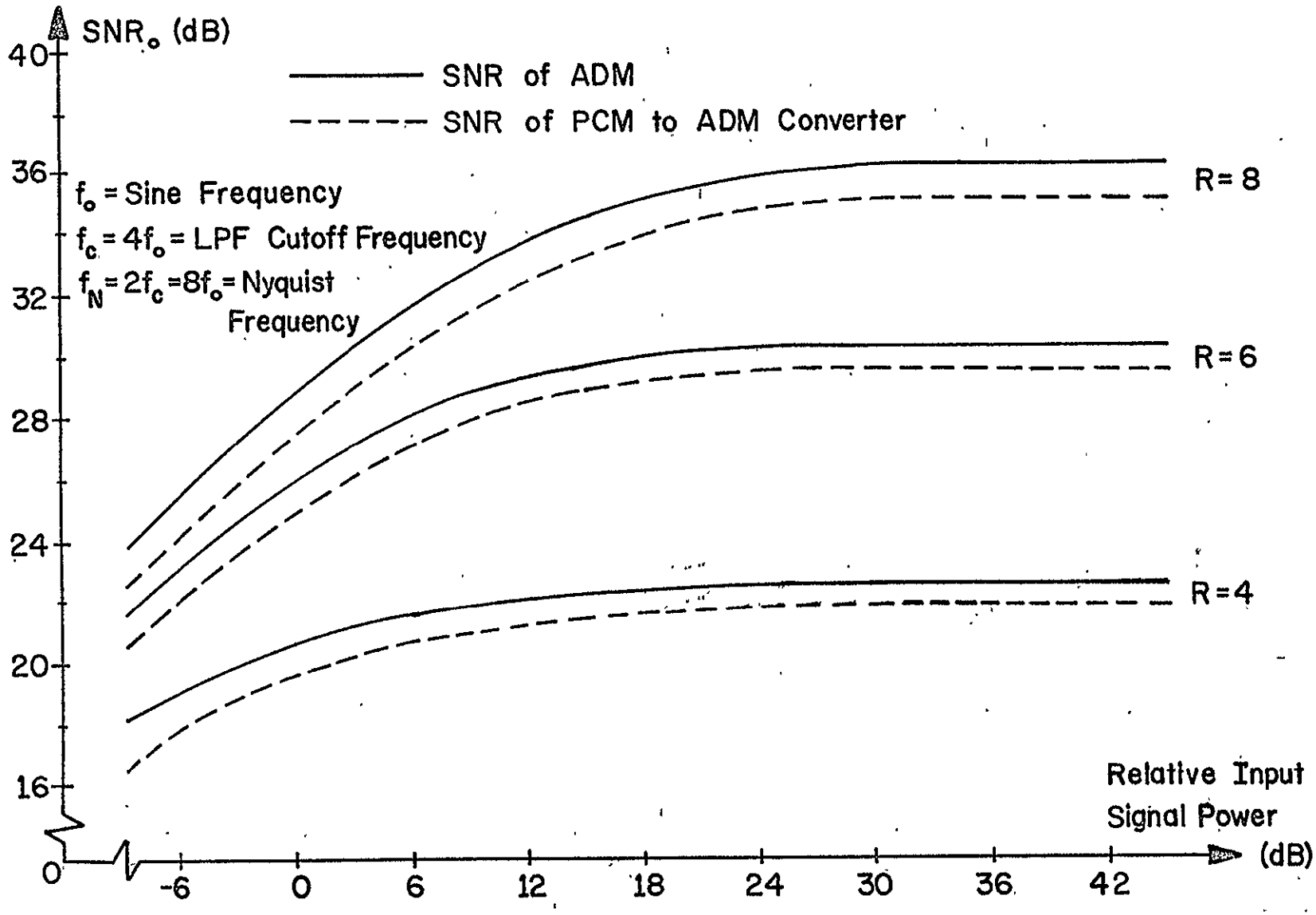


Fig. 4.3-10. SNR of PCM to ADM Converter with Low Frequency Tone

estimate points between PCM samples, when a low frequency tone is used as the input. The input sinusoid is of frequency f_o; the LPF cutoff frequency is fixed at $4f_o$; and the Nyquist rate set at $8f_o$. On this graph, we show SNR_o versus relative input signal power, over a range of 54 dB, when the parameter $R = f_s/f_N$ takes on values 4, 6 and 8. To have a basis of comparison, we give the SNR curves obtained from an ADM, operating at a rate f_s, when the input is a sinusoidal signal. This represents the best performance that can be achieved when an ADM is used in the converter. All these plots are smooth curves drawn from the mean SNR_o and an average standard deviation of 1-2 dB. We observe, as expected, that this converter works very well for a low frequency signal.

When employing the high frequency tone as an input, i.e., frequency $3f_o$, we maintained the same LPF cutoff frequency and the same PCM rate. Likewise, the input signal was varied over the same range, from an amplitude of 5 minimum step sizes at -6 dB to 1280 minimum step sizes at 42 dB. The output SNR is shown in Fig. 4.3-11 for the basic PCM to ADM converter and an ADM encoder. The common parameter used is R = 4 and 8. We have again plotted the average SNR_o, with a standard deviation of approximately 1-3 dB, to obtain smooth curves for both the converter and the encoder.

For the low frequency tone, the performance is quite acceptable since it is within about 1 dB of the maximum SNR. However, we find the figure of merit down 4 and 8 dB, for

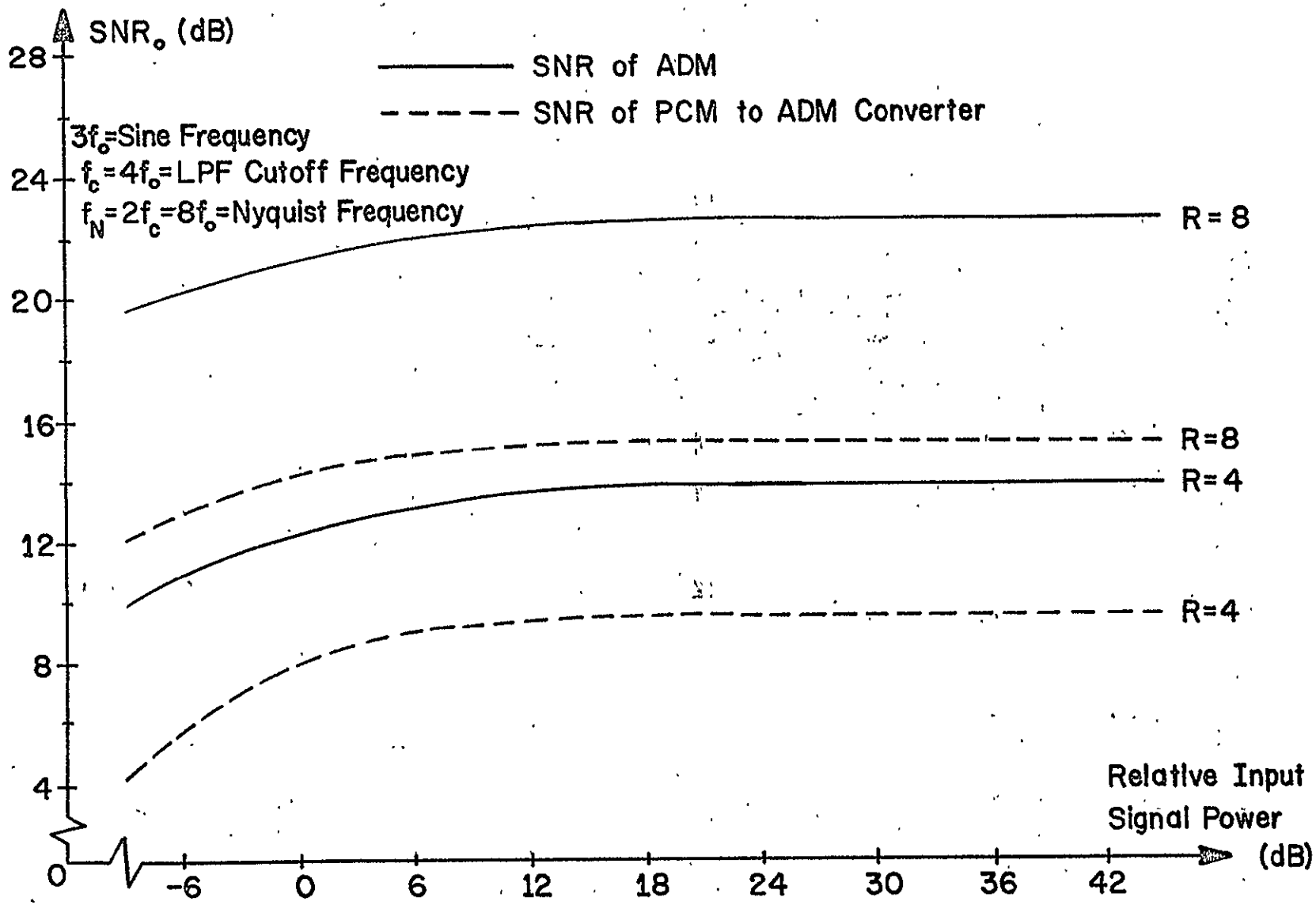


Fig. 4.3-11. SNR of PCM to ADM Converter with High Frequency Tone

R = 4 and 8, respectively, from the best SNR obtainable. Consequently, the basic converter is not acceptable in its present form and we must at least add a midpoint PCM estimate to raise the performance to reasonable quality. In the next section, we present a non-parametric estimation technique that will do just this.

4.4 Non-Parametric PCM-ADM Converter

The obvious advantage of designing a non-parametric PCM to ADM converter is that the structure of the system is independent of the input signal. We do not have to know the probability statistics or the power spectrum of the input. In considering the design of a non-parametric converter, we still wish to maintain a simple structure that can be readily built with all-digital hardware. In addition, we must avoid both the "multipath" and the "endpoint" problems discussed in Sec. 4.1.

4.4.1 PCM Estimation Technique

Let us consider the structure of the converter discussed in Sec. 4.3. With the ADM encoder as part of the converter, we automatically eliminate the "multipath" and "endpoint" problems. Thus, we must only deal with the derivation of $v(k)$, i.e., the input to the ADM, from the PCM samples, $x(Rk)$. As a natural extension of the technique described in Sec. 4.3, we can calculate the ADM input values using a method similar to that used to estimate points between

adjacent PCM samples. Taking this to its limit, we can determine R-1 estimate points every T_N seconds.

Deriving the estimate points from the PCM samples can be viewed as a filtering operation to obtain each estimate. A non-recursive digital filter structure would yield a specific weighting for every estimate point. Certainly, filters are designed without knowledge of the input signal statistics or its power spectrum. It thus becomes a question of what non-parametric design technique is optimal to employ when calculating estimate points between PCM samples. In the next section, we shall describe the best weighting technique that requires knowledge of only the bandwidth of the system.

4.4.2 Ideal LPF Impulse Response Weighting

If we consider the PCM points to be impulses, then the optimum technique to recover the input signal would be to ideal LPF the impulses. Therefore, we can find the value of the signal at any point between PCM samples by a superposition of the input impulses. This concept is one statement of the sampling theorem [34], that is, if x(t) is bandlimited to f_c then it can be represented by the equation

$$x(t) = \sum_{k=-\infty}^{\infty} x(kT_N)h(t - kT_N) \ , \qquad (4.4\text{-}1)$$

where

$$T_N = 1/2f_c \ . \qquad (4.4\text{-}2)$$

The "interpolation function",

$$h(t) \equiv \frac{\sin 2\pi f_c t}{2\pi f_c t} , \qquad (4.4\text{-}3)$$

is the impulse response of an ideal rectangular filter with transfer function

$$H(f) \equiv \begin{cases} T_N , & |f| < f_c , \\ 0 , & |f| > f_c . \end{cases} \qquad (4.4\text{-}4)$$

The terms $x(kT_N)$ are the PCM sample values, previously denoted as $x(Rk)$. To exactly calculate points between adjacent PCM samples will take an infinite number of samples, thus infinite time.

The form of the estimator given in Eq. (4.4-1) is exactly the structure of a digital recursive filter. Therefore, the implementation is readily adaptable to digital hardware. However, there are practical considerations that must be addressed. We must truncate the sum in Eq. (4.4-1) to a finite number of terms and still maintain an accurate estimate of the signal value. If we eliminate some of the R-1 estimate points and use instead the staircase approximation originally presented in Sec. 4.3, the complexity of the converter can be considerably reduced. We wish to know how many estimate points can be omitted and still achieve acceptable performance. These practical considerations are dealt with by reverting back to the basic PCM to ADM converter that employs only one estimate point between PCM samples.

4.4.3 Converter with Midpoint Estimate

Let us first consider the truncation of the infinite sum given in Eq. (4.4-1). The most general solution to this problem is by application of the Landau-Pollak bandwidth-constraint theorem [35]. However, this essentially requires that we consider the sampling function, h(t), to be negligible after a certain value of t. The Landau-Pollak theorem gives the error that will result from a finite number of terms in the estimation equation. We have found, with various simulation signals, that if h(t) is neglected after its second zero, then the estimate obtained with four adjacent PCM samples is within 10% of the true signal value. This will act as a "rule of thumb" for the accuracy of our estimate points since the use of more PCM samples in the estimator does not significantly improve the performance of this converter.

To determine the number of estimate values that can be eliminated and replaced by a staircase waveform, we return to the PCM to ADM converter with midpoint estimate. Using the ideal LPF weighting function with four adjacent PCM samples, we calculate the midpoint with the following equation:

$$\begin{aligned} x'(Rk + R/2) = {} & h(3T_N/2)x(Rk - R) \\ & + h(T_N/2)x(Rk) \\ & + h(-T_N/2)x(Rk + R) \\ & + h(-3T_N/2)x(Rk + 2R) \ . \end{aligned} \tag{4.4-5}$$

To show the qualitative effect of such a system, we have determined the midpoints needed in a typical high tone test signal using the above algorithm for $x'(Rk + R/2)$. By observing the straight line approximation, $w(t)$, superimposed on the test tone in Fig. 4.4-1, we see that this represents a fairly good approximation to the input sinusoid.

We shall see that the performance of this converter comes within about 1 dB of the optimum figure of merit curve. In Fig. 4.4-2, we show a block diagram of the PCM to ADM converter with midpoint estimates using four adjacent PCM points. The weighting coefficients are defined as

$$h_1 = h(\pm 3T_N/2) = 2/3\pi \qquad (4.4\text{-}6)$$

and

$$h_2 = h(\pm T_N/2) = 2/\pi \ . \qquad (4.4\text{-}7)$$

This system represents a simple, straightforward hardware realization. To improve the converter performance and come closer than 1 dB to the optimum SNR curve by estimating 2 or 3 points between PCM samples, it would be necessary to increase the hardware of the system by about a factor of two. An unrewarding tradeoff for less than 1 dB increase in output SNR.

4.4.4 Simulation and Performance

Using the PDP 8/L, the PCM to ADM converter depicted in Fig. 4.4-2 was simulated and tested with elementary input signals to verify its operation. Similar to the technique

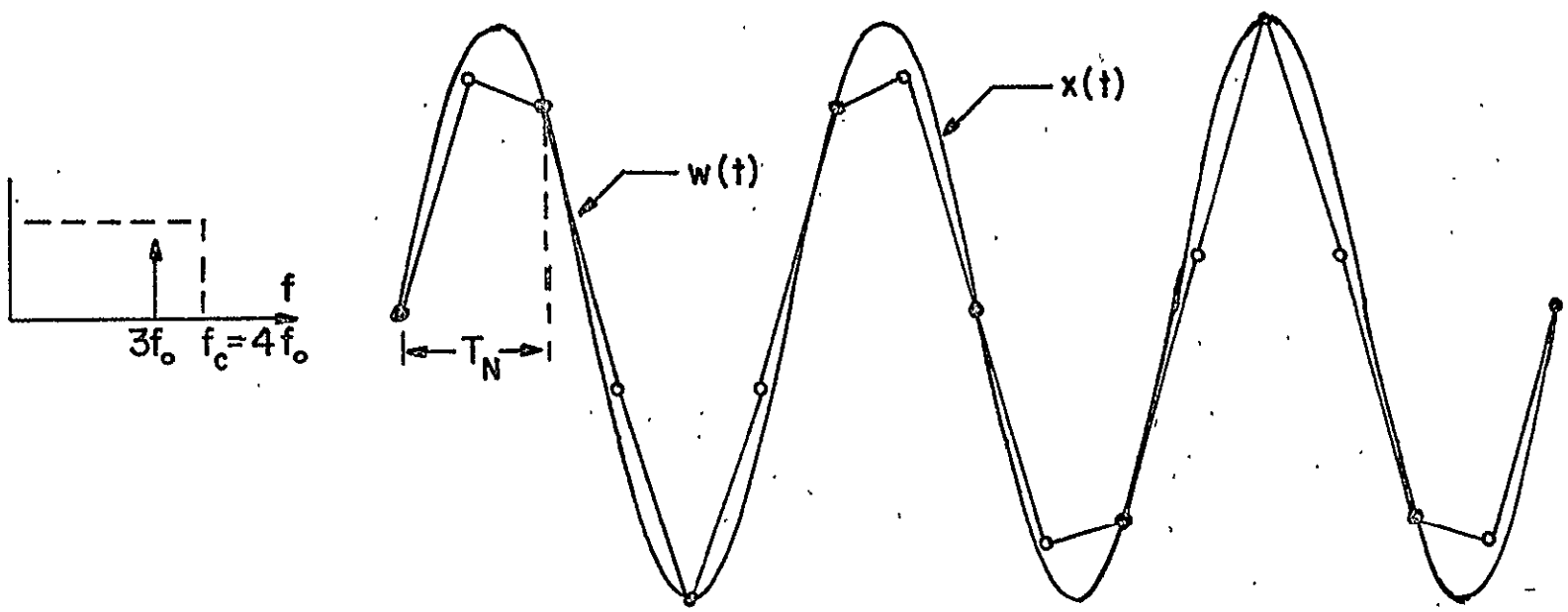


Fig. 4.4-1. Straight Line Approximation between PCM Points and Non-Parametric PCM Midpoint Obtained from Four Adjacent PCM Samples

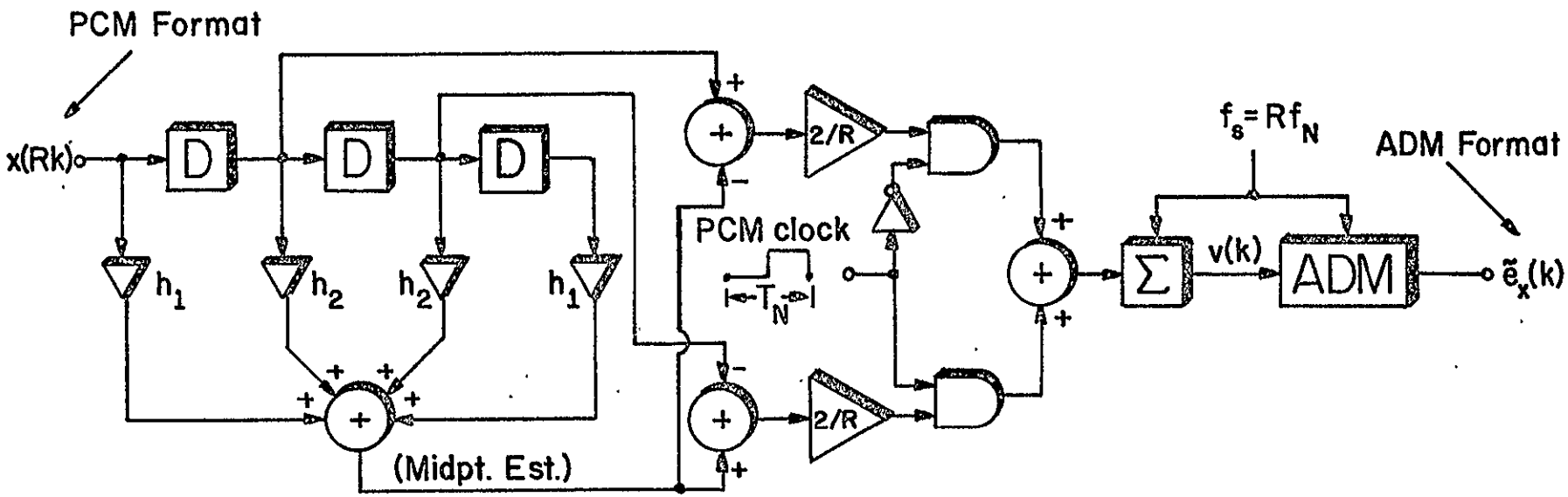


Fig. 4.4-2. Non-Parametric PCM to ADM Converter with Midpoint Estimate using Four Adjacent PCM Samples

employed in Ch. 3, the coefficients h_1 and h_2 were approximated by the nearest integral multiple of 1/16. The resulting decrease in output SNR was insignificant to warrant more precision in realizing the scalars h_1 and h_2. To determine the performance of this device, we repeat the procedure employed in Sec. 4.3 with the high frequency tone. The simulation system is exactly as shown in Fig. 4.3-7. We calculate the Fourier series representation of the ADM estimate, $\hat{v}(k)$, and determine the output SNR after low pass filtering $\hat{v}(k)$ and truncating the noise power. The SNR acts as a random variable dependent upon the starting point of the input sinusoid and we must calculate its mean and standard deviation. This is all done only for the high frequency tone since the basic PCM to ADM converter performed very well with the low frequency input.

The performance curves given in Fig. 4.4-3 are smooth plots of the average SNR_o, with a standard deviation of 1-2 dB, over a relative input power range of 54 dB. The parameter R takes on values of 4 and 8 yielding a system that makes 2 or 4 steps between each PCM sample and its adjacent midpoints. The graph in Fig. 4.4-3 represents an extremely comprehensive view of the performance of this converter. As before, we show the SNR of the PCM to ADM converter with zero estimate points, i.e., the basic converter. We have plotted the SNR when one estimate point, the midpoint, is non-parametrically evaluated using four adjacent PCM samples. Observe that there is an increase of 7 and 3 dB

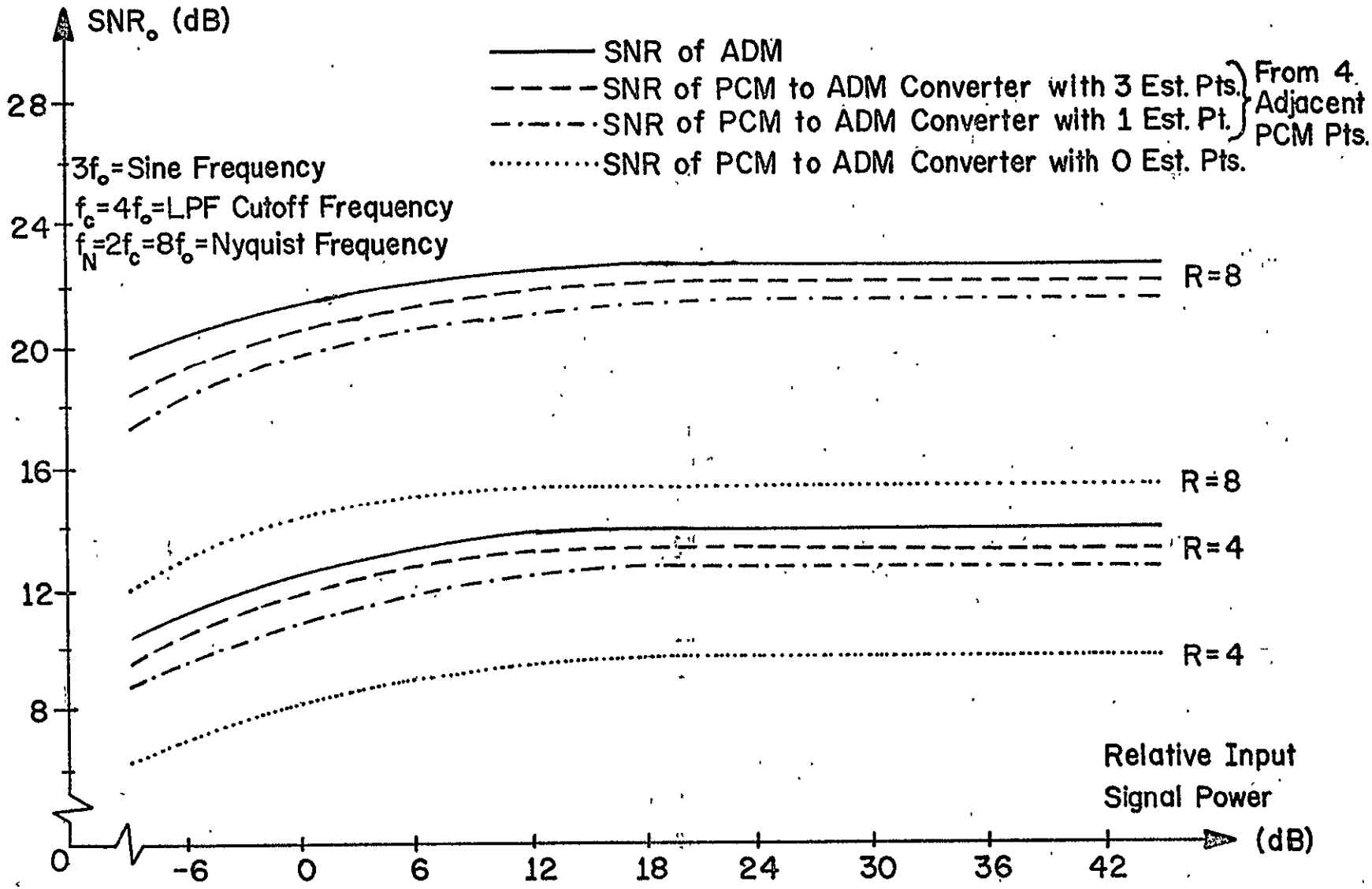


Fig. 4.4-3. SNR of Non-Parametric PCM to ADM Converters with High Frequency Tone

over the basic converter when R = 8 and 4, respectively. From the last two curves, we obtain a good feeling for the worth of the converter with midpoint estimate. When three estimate points are derived from four adjacent PCM samples, there is only an increase of 0.5 dB. Likewise, the optimum performance, when we ADM encode samples at f_s, yields only a 1 dB increase over the system given in Fig. 4.2-2. We cannot use two estimate points between PCM samples because we must have an integral number of steps between the various PCM values. With the values of R utilized, this is impossible.

We can conclude that the non-parametric PCM to ADM converter, with midpoint estimate, is a rather successful system. It represents a compromise between a minimum amount of very simple digital hardware and the maximum figure of merit. Since there is so little improvement in performance by going to a converter which utilizes three estimate values between PCM points, we are quite content not to extend the complexity of this device. Furthermore, with present-day technology, this entire system could be constructed, using large-scale integration, on one integrated circuit chip.

4.5 Other PCM-ADM Converters

Following the converter concept presented previously, where a digital filter operates on the PCM samples to produce signal estimates, at a rate f_s, which are then ADM encoded, we can introduce several other PCM to ADM converters. The essential difference is the technique that is employed

to derive the signal estimates, i.e., the digital filter. We shall consider digital filters which range from an extremely simple estimator circuit to a submultiple sampling filter designed with a Z-transform approach.

4.5.1 Digital Zero-Order Hold Circuit

Perhaps the simplest device that can be employed to estimate values from the PCM samples is a zero-order hold filter. This technique is analogous to the concept used by D. Goodman and J. Flanagan [36] to convert from ADM to linear DM formats. A zero-order hold circuit will keep the same PCM value for R ADM periods every Nyquist interval. This PCM value is used as the input to an ADM and the encoder treats it like a step input. Thus, the ADM estimate approaches the PCM level and then hovers about it for the remainder of the Nyquist interval.

We could, of course, simply use a D/A converter and a holding circuit to keep the input at the ADM constant for the entire Nyquist interval. However, we would like to employ a completely digital hold circuit. If we define the digital zero-order hold filter as a circuit which holds one value at its output for R periods when the held value is the input during the first period and there is zero input for the second through R^{th} periods, then it has the following transfer function:

$$G_o(z) = \sum_{i=0}^{R-1} z^{-i} , \qquad (4.5\text{-}1)$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

where z^{-1} represents a delay of T_s seconds. From the block diagram shown in Fig. 4.5-1, it is obvious that this circuit will produce the desired result.

An alternate digital hold circuit would be a bank of flip-flops, all operating on the PCM clock, to store the PCM sample word and hold it at the output until a new PCM sample was clocked into the flip-flops. This realization, although not giving rise to a more mathematical transfer function, as Eq. (4.5-1), is the practical approach to the problem because of its hardware simplicity. Although we will not analyze this system in depth, we shall make some pertinent comments about its operation. We can determine the exact effect that the digital hold circuit will have on x(Rk) by evaluating $G_o(\omega)$ from Eq. (4.5-1) with

$$z = \exp(j\omega T_s) \quad . \tag{4.5-2}$$

However, for the purposes of this discussion, we can simply recall that an analog holding circuit transfer function, given in Eq. (3.3-29),represents a crude low pass filter.

Viewing the operation of this converter in the frequency domain, we wish to eliminate the high frequency components found in the sampled x(t) signal and retain, undisturbed, the baseband spectrum. But, we expect the performance of this converter to be worse than the basic converter described in Sec. 4.3. The basic converter employed a filter which produced a straight line between PCM samples and used values on this line as the input to the ADM. This filter is

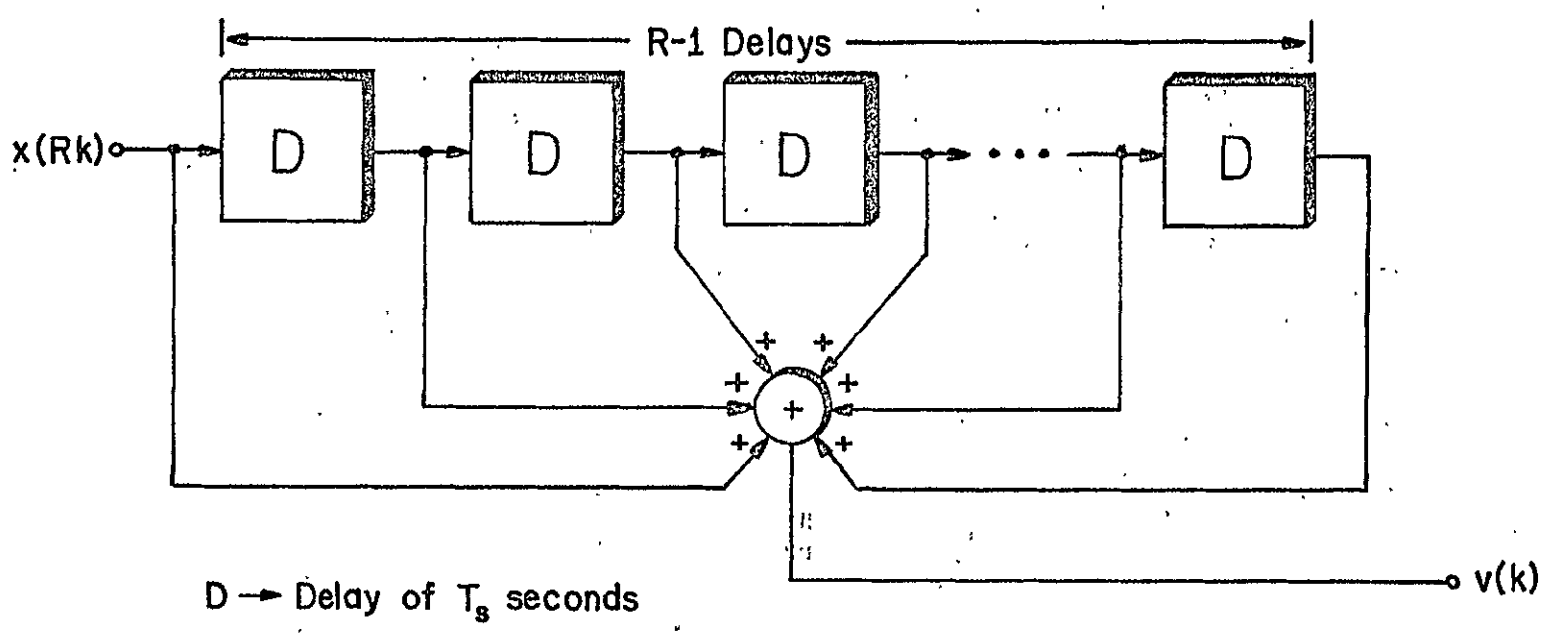


Fig. 4.5-1. Digital Hold Circuit

a first-order hold and acts as a better LPF than the zero-order hold circuit. The only real virtue of the zero-order hold converter is its utter simplicity.

4.5.2 Submultiple Sampling Technique

A more refined approach to PCM to ADM conversion is to actually consider the design of a digital LPF which extracts signal estimates at a rate f_s from the PCM samples, which occurs at a rate $f_N = f_s/R$. The digital filter that will perform this function must operate in the submultiple sampling mode [37]. When a digital system operates in the submultiple sampling mode it simply means that the system produces outputs at a frequency which is an integral multiple of the input frequency. This is exactly the case that exists in this converter. To thoroughly describe the submultiple sampling technique, we shall introduce two possible system designs.

The first system design uses the PCM samples as inputs to the filter with R-1 zeros inserted between PCM samples. Since we shall explain this theory in a general fashion, omitting the particular characteristics of the LPC, let us start with the Z-transform of a digital LPF represented by $H_s(z)$ and the input and output of the filter specified by $X_i(z)$ and $X_o(z)$, respectively. Then, as we normally expect,

$$X_o(z) = H_s(z)X_i(z) \quad , \qquad (4.5\text{-}3)$$

where z^{-1} now represents a delay of T_N seconds.

If we want the output to occur R times in the Nyquist sampling period and the input to have R-1 zeros between the PCM samples of $X_i(z)$, then the output is expressed as

$$X_o(z)_R = H_s(z)_R X_i(z) \quad , \tag{4.5-4}$$

where

$$H_s(z)_R = H_s(z) \Big|_{\substack{z = z^{1/R} \\ T_N = T_N/R}} \quad . \tag{4.5-5}$$

$H_s(z)_R$ is referred to as the submultiple sampling Z-transform of the digital LPF, $H_s(z)$, and is realized exactly the same as the ordinary digital filter except delay elements are reduced by a factor R and scalars with the constant T_N included are attenuated by the factor R. This implies that the digital filter operates at a frequency Rf_N. One out of every R cycles sets the input to a PCM sample and the rest are inserted zero. The filter operates on these inputs to produce signal estimates which are used as the input to the ADM.

The second system design employs the PCM samples as filter inputs for R ADM periods every T_N seconds. This is similar to the PCM samples being held for the Nyquist period. If we incorporate the digital zero-order hold circuit presented above with the submultiple sampling mode digital LPF, then we shall have achieved the desired design system. The PCM-held filtered output is given by

$$X_{o_H}(z)_R = G_o(z)_R H_s(z)_R X_i(z) \quad , \tag{4.5-6}$$

where

$$G_o(z)_R = \sum_{i=0}^{R-1} z^{-i/R} \quad . \qquad (4.5\text{-}7)$$

Again, the filter outputs are used as the input to the ADM to complete the conversion.

The actual characteristics of the digital LPF and the comparison of these two design techniques is left for future research. We expect that the performance will be directly proportional to the order of the digital LPF chosen. Inherently, it seems that the holding circuit should add an additional low pass filtering effect and thereby enhance the system operation. These conjectures are left to be verified by a more thorough investigation.

4.5.3 Recent Developments

There is not a great deal of literature that deals with systems to perform PCM to DM conversion and the related difficulties involved with such systems. Some very brief articles employ a binary rate multiplier and digital filters in converting from PCM to linear DM [38]. These studies are geared toward extremely simple hardware for the purpose of inexpensive realization. However, they must employ a tremendously high bit rate for the linear DM, in the order of several megabits per second.

Virtually nothing has been reported that seriously considers PCM to ADM conversion. There is no practical, systems-oriented research nor do we find any detailed theoreti-

cal analysis. Therefore, this area is almost completely open for many new ideas and much further investigation.

CHAPTER 5

CONCLUSIONS

In this final chapter, we summarize the contributions of this research work. We also present a brief comparison between the systems designed and other digital circuits, where practical, or optimal systems which perform the same function. In addition, we discuss new problems that can be addressed as natural extensions of this thesis.

The work done on the topics considered in this thesis is viewed as a success. We were able to solve the three problems introduced with simple, all-digital systems and achieve good performance with simple hardware. In Ch. 2, we have conclusively shown that arithmetic processing can be performed on ADM encoded signals by operation on the ADM bit streams. Thus, we do not have to resort to PCM or analog processors. We have demonstrated that the performance and hardware complexity of the direct ADM adder and multiplier are comparable to those of equivalent PCM devices.

In Ch. 3, we have derived a theoretical design technique for constructing an ADM to PCM converter. We have also shown that, for moderate ADM bit rates, the SNR of this system is within 1 dB of the performance of the ideal analog demodulation converter. Neither of these two accomplishments is, however, the most important aspect of this work. The most significant contribution is the thorough explanation of why

some sort of LPF is needed in converting from ADM to PCM formats.

The work on PCM to ADM conversion addresses a problem almost untouched in the literature and obtains significant, practical results for the first time. We refer particularly to the "multipath" problem and the various ways that it can be solved, i.e.: with probabalistic statistics of the input signal, or parametrically, utilizing the input power spectrum, or non-parametrically, using no information about the input signal except its bandwidth. The SNR obtained for the non-parametric PCM to ADM converter, with one estimate PCM point obtained from four adjacent PCM samples, came within 1 dB of the optimal PCM to ADM converter, in which samples of the input signal are directly ADM encoded.

From the basic results of these three chapters, we see that the overall purpose of this research was accomplished. We were able to design direct ADM arithmetic processors and to devise conversion systems between ADM and PCM formats with simple, all-digital circuits that yield good performance. This leads us to the additional ideas that were generated by this thesis and new work that can be done as an extension of it.

Other than the few examples mentioned in Ch. 2, the use and applications of the ADM arithmetic processors still remain to be explored. This is also true for the conversion systems, particularly PCM to ADM since, ADM to PCM conversion can generally be employed as an intermediate step in A/D con-

version. There are two new ideas that emerge from the study of direct arithmetic processing of ADM encoded signals. The first concept is the mapping of the ADM bits, $e_x(k)$ and $e_y(k)$, into a processed ADM bit, $e_r(k)$. The processed bit can have 2, 3 or more values depending on the current and perhaps even on the past ADM input bits. This processed bit would then be employed in an algorithm, let us say, similar to that of a Song audio mode decoder to generate an estimate of the sum or product of the two input signals. The mapping of $e_x(k)$ and $e_y(k)$ into $e_r(k)$ as well as perhaps the decoding algorithm for $e_r(k)$ will be different for addition and multiplication. The second idea, completely different from the first, is to convert the ADM encoded signals to linear DM formats, operating at a higher rate. We can now use the very simple linear DM multiplier, designed to obtain the product, and then convert back to ADM. The considerations in this approach are performance and the hardware complexity required to construct such a system.

In Chs. 3 and 4, we have briefly mentioned several other conversion techniques that require a follow-up study. In particular, the submultiple sampling filter approach to PCM to ADM conversion appears to be a rather easy and straightforward solution to this problem. We must consider the effect of starting with a companded rather than a linear PCM format. We would like to investigate the variation in performance as a function of the order and type of filter employed. We can employ different ways of measuring converter

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

performance. If we use white Gaussian noise, bandlimited from 500 to 1000 Hz, to represent a speech waveform, then we can encode this waveform in ADM or PCM and apply our conversion process. We can then measure output power in and out of the 500 Hz band to determine converter performance. This represents, a possible, significant improvement over the results obtained with a single test tone. Likewise, we could generate voice tapes with systems that have been physically realized.

This brings us to the last few thoughts that could be pursued as future work. For all three topics discussed, since the designs have been oriented toward digital hardware, we would like to construct real time versions of the processors and converters. Then, we can truly test voice signals with subjective evaluation and even extend all the work to the Song video mode ADM and observe the effect on picture quality. An experimental study of this type will bring out additional practical aspects of the work accomplished here, some of which may not appear otherwise.

APPENDIX 1

AMPLITUDE-FREQUENCY CHARACTERISTICS OF THE FOUR-TERM NON-RECURSIVE AVERAGING FILTER

In this appendix, we derive the expression for $|H_a(\omega)|$, i.e., Eq. (2.3-9), from the digital transfer function,

$$H_a(z) = (1 + z^{-1} + z^{-2} + z^{-3})/4 \ . \qquad \text{(A1-1)}$$

If we set

$$z = \exp(j\omega T_s) \qquad \text{(A1-2)}$$

and separate terms, we obtain

$$H_a(\omega) = (1 + e^{-j\omega T_s})/4 + e^{-j2\omega T_s}(1 + e^{-j\omega T_s})/4 \ . \qquad \text{(A1-3)}$$

Factoring Eq. (A1-3), it reduces to

$$H_a(\omega) = (1 + e^{-j\omega T_s})(1 + e^{-j2\omega T_s})/4 \ . \qquad \text{(A1-4)}$$

By removing $e^{-j\omega T_s/2}$ from the first term of this product and $e^{-j\omega T_s}$ from the second term, we reveal a rather familiar form

$$H_a(\omega) = e^{-j\omega T_s/2}[(e^{j\omega T_s/2} + e^{-j\omega T_s/2})/2] \qquad \text{(A1-5)}$$

$$\cdot\ e^{-j\omega T_s}[(e^{j\omega T_s} + e^{-j\omega T_s})/2] \ . \qquad \text{(A1-5)}$$

Recognizing the exponential definition of the cosine, we can now write

$$H_a(\omega) = e^{-j3\omega T_s/2}\cos(\omega T_s/2)\cos(\omega T_s) \quad . \qquad \text{(A1-6)}$$

Thus, by observation, we see that the amplitude-frequency characteristics of the four-term non-recursive averaging filter are given by

$$|H_a(\omega)| = |\cos(\omega T_s/2)\cos(\omega T_s)| \quad . \qquad \text{(A1-7)}$$

APPENDIX 2

THE PCM PRODUCT VIA ALTERNATE TECHNIQUES

There are many other techniques to obtain the PCM product of x(t) and y(t) by using the samples of both signals at a rate $2f_m$. Some of these shall be discussed in this appendix. It is clear that we can low pass filter the samples of each signal, return to the analog domain, perform an analog multiplication and then resample at $4f_m$. This will certainly give us PCM samples of the product. However, we must resort to a complete analog multiplication. This can be avoided by holding the samples of each signal for $1/2f_m$ seconds and multiplying the held values. If we assume that the samples are actually quantized values, then we have a multiplication of discrete values yielding a discrete product--just like digital multiplication. The only difference is that the product is a held waveform rather than being samples. To obtain the PCM product, we must resample the held product at $4f_m$ and A/D convert.

We shall show first that the holding and multiplying technique described above does, in fact, produce the product of x(t) and y(t) and second that it is not a good technique to employ because of the distortion that it introduces into the product. It is intuitively apparent that this technique will yield the product because, by holding the samples of each signal, we are low pass filtering them. The transfer

function of the holding circuit for a signal bandlimited to f_m and sampled at the Nyquist frequency is given as [39]

$$G_H(\omega) = \frac{\sin(\omega T_N/2)}{\omega T_N/2} , \qquad \text{(A2-1)}$$

where

$$T_N = 1/2f_m \equiv \text{the Nyquist period.} \qquad \text{(A2-2)}$$

Although the holding circuit has LPF characteristics, they are not very flat in-band and do not cutoff sharply. This causes distortion of the in-band signal spectrum and does not completely eliminate the out-of-band spectrum.

To demonstrate this concept, we define the Fourier transform of x(t) and y(t) as

$$X(\omega) \equiv \mathfrak{F}[x(t)] \qquad \text{(A2-3)}$$

and

$$Y(\omega) \equiv \mathfrak{F}[y(t)] . \qquad \text{(A2-4)}$$

The sampled spectra of x(t) and y(t) are given as

$$X^*(\omega) = \sum_{n=-\infty}^{\infty} X(\omega + n\omega_N) \qquad \text{(A2-5)}$$

and

$$Y^*(\omega) = \sum_{n=-\infty}^{\infty} Y(\omega + n\omega_N) \qquad \text{(A2-6)}$$

where

$$\omega_N = 2\pi f_N \qquad \text{(A2-7)}$$

and

$$f_N = 1/T_N \equiv \text{the Nyquist frequency.} \qquad \text{(A2-8)}$$

After holding, $X^*(\omega)$ and $Y^*(\omega)$ are multiplied by $G_H(\omega)$ which greatly attenuates, but does not completely eliminate, the out-of-band spectra. If we multiply the held signals in the time domain, we must convolve their frequency domain spectra. Denoting this product spectrum as $Z(\omega)$, it is expressable as

$$Z(\omega) = \int_{-\infty}^{\infty} G_H(\nu)X^*(\nu)G_H(\omega - \nu)Y^*(\omega - \nu)d\nu \ . \qquad \text{(A2-9)}$$

To show that the actual product spectrum is imbedded in $Z(\omega)$, we must decompose $X^*(\omega)$ and $Y^*(\omega)$ into its in-band and out-of-band spectra, i.e.,

$$X^*(\omega) = X(\omega) + X_o(\omega) \qquad \text{(A2-10)}$$

where

$$X_o(\omega) = \sum_{n=1}^{\infty} [X(\omega + n\omega_N) + X(\omega - n\omega_N)] \qquad \text{(A2-11)}$$

and

$$Y^*(\omega) = Y(\omega) + Y_o(\omega) \qquad \text{(A2-12)}$$

where

$$Y_o(\omega) = \sum_{n=1}^{\infty} [Y(\omega + n\omega_N) + Y(\omega - n\omega_N)] \ . \qquad \text{(A2-13)}$$

Now we can recast $Z(\omega)$ in the following form:

$$Z(\omega) + \int_{-\infty}^{\infty} G_H(\nu)G_H(\omega - \nu)X(\nu)Y(\omega - \nu)d\nu + Z_{dl}(\omega) \ , \qquad \text{(A2-14)}$$

where

$$Z_{dl}(\omega) = \int_{-\infty}^{\infty} G_H(\nu)G_H(\omega - \nu)[X(\nu)Y_o(\omega - \nu) + X_o(\nu)Y_o(\omega - \nu)]d\nu \qquad \text{(A2-15)}$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

and this term, $Z_{d1}(\omega)$, represents distortion.

With the aid of some trigonometric identities, we can expand $G_H(\nu)G_H(\omega - \nu)$ into the following form:

$$G_H(\nu)G_H(\omega - \nu) = \frac{\sin(\omega T_N/2)\sin(\nu T_N/2)\cos(\nu T_N/2)}{(\omega\nu - \nu^2)T_N^2/4} - \frac{\cos(\omega T_N/2)\sin^2(\nu T_N/2)}{(\omega\nu - \nu^2)T_N^2/4} . \tag{A2-16}$$

Letting the second term of Eq. (A2-16) also contribute to distortion, we rewrite $Z(\omega)$ as

$$Z(\omega) = \frac{\sin(\omega T_N/2)}{T_N^2/4} \int_{-\infty}^{\infty} \frac{\sin(\nu T_N)}{\omega\nu - \nu^2} X(\nu)Y(\omega - \nu)d\nu + Z_{d2}(\omega) + Z_{d1}(\omega) . \tag{A2-17}$$

where

$$Z_{d2}(\omega) = \frac{-\cos(\omega T_N/2)}{T_N^2/4} \int_{-\infty}^{\infty} \frac{\sin(\nu T_N/2)}{\omega\nu - \nu^2} X(\nu)Y(\omega - \nu)d\nu . \tag{A2-18}$$

By expanding $\sin(\nu T_N)$ into its power series,

$$\sin(\nu T_N) = \nu T_N - (\nu T_N)^3/3! + (\nu T_N)^5/5! - + \cdots , \tag{A2-19}$$

canceling the common factor (ν) from the denominator $(\omega\nu - \nu^2)$, and then writing the power series for $1/(\omega - \nu)$ as

$$(1/\omega)[1/(1 - \nu/\omega] = (1/\omega)[1 + \nu/\omega + (\nu/\omega)^2 + \cdots] \ , \tag{A2-20}$$

we can express $\sin(\nu T_N)/(\omega\nu - \nu^2)$ as

$$\sin(\nu T_N)/(\omega\nu - \nu^2) = (T_N/\omega)[1 + \nu/\omega + (\nu/\omega)^2 + \cdots]$$
$$- (T_N^3/\omega 3!)[\nu^2 + \nu^3/\omega + \nu^4/\omega + \cdots] + - \cdots \ . \tag{A2-21}$$

If we allow all terms but T_N/ω in Eq. (A2-21) to contribute to distortion, then we can write $Z(\omega)$ as:

$$Z(\omega) = \frac{4\sin(\omega T_N/2)}{\omega T_N/2} \int_{-\infty}^{\infty} X(\nu)Y(\omega - \nu)d\nu + Z_{d3}(\omega)$$
$$+ Z_{d2}(\omega) + Z_{d1}(\omega) \ , \tag{A2-22}$$

where

$$Z_{d3}(\omega) = \frac{\sin(\omega T_N/2)}{\omega T_N/2} \int_{-\infty}^{\infty} \left(\frac{\sin(\nu T_N)}{\omega\nu - \nu^2} - \frac{T_N}{\omega} \right) X(\nu)Y(\omega - \nu)d\nu \ . \tag{A2-23}$$

We recognize the spectrum of the product in Eq. (A2-22) as the convolution of $X(\omega)$ and $Y(\omega)$. However, we observe that it is shaped by $\sin(\omega T_N/2)/(\omega T_N/2)$ and there are the additive distortion terms, $Z_{d1}(\omega)$, $Z_{d2}(\omega)$ and $Z_{d3}(\omega)$. Thus, although we have shown that the actual product is embedded in $Z(\omega)$, we have also shown that we do not expect the product to be of very high quality because of all the distortion contained in $Z(\omega)$.

APPENDIX 3

FREQUENCY CHARACTERISTICS OF THE CASCADE ARRANGEMENT FILTER, $H_{CD}(z)$, USED IN ADM TO PCM CONVERSION

In this appendix, we present a derivation of the transfer function, $H_{CD}(f)$, given in Eq. (3.3-26), which shows the effect of the non-recursive filter (Fig. 3.3-2) on the ADM estimate, $\hat{x}(k)$. The filter coefficients, $g(i)$, are obtained from the step response of an ILPF (Fig. 3.3-5). We have set Q, the number of filter coefficients, to be an even integer and specified $g(0) = 0$. The remaining Q-1 coefficients are symmetrically distributed about 0.5 on the $g(t)$ curve. Due to the symmetry of $g(t)$, this implies that

$$g(Q/2) = 0.5 \ , \tag{A3-1}$$

$$g(i) = 0.5 - g'(i), \text{ for } i = 1, 2, \cdots, Q/2 - 1, \tag{A3-2}$$

and

$$g(i) = 0.5 + g'(Q - i), \text{ for } i = Q/2 + 1, Q/2 + 2, \cdots, Q - 1 \ . \tag{A3-3}$$

To find the frequency characteristics of $H_{CD}(z)$, where

$$H_{CD}(z) = z^{-Q} + (1 - z^{-1}) \sum_{i=0}^{Q-1} g(i) z^{-1} \ , \tag{A3-4}$$

we first evaluate the frequency spectrum of

$$H_o(z) = \sum_{i=0}^{Q-1} g(i) z^{-i} \ . \qquad \text{(A3-5)}$$

Factoring out $z^{-Q/2}$ and noting that $g(0) = 0$, Eq. (A3-5) becomes

$$H_o(z) = z^{-Q/2} \sum_{i=1}^{Q-1} g(i) z^{Q/2-i} \qquad \text{(A3-6)}$$

If we substitute Eqs. (A3-1), (A3-2) and (A3-3) in the above expression for $H_o(z)$ and gather terms from the sum, in pairs of two, by mating the terms $(i = 1, i = Q - 1)$, $(i = 2, i = Q - 2)$, $\cdots$, we shall produce $Q/2 - 1$ pairs of the form

$$0.5[z^{Q/2 - m} + z^{-(Q/2 - m)}] - g'(m)[z^{Q/2 - m} - z^{-(Q/2 - m)}] \ ,$$

where

$$m = 1, 2, \cdots, Q/2 - 1 \ ,$$

and one term, when $i = Q/2$, yielding $g(Q/2)z^o = 0.5$.

By letting

$$z = \exp(j\omega T_s) \ , \qquad \text{(A3-7)}$$

we see that each pair gives rise to a cosine and a sine term of the form

$$\cos(\omega(Q/2 - m)T_s) - 2jg'(m)\sin(\omega(Q/2 - m)T_s) \ .$$

If we now set

$$m = Q/2 - i \ , \qquad \text{(A3-8)}$$

sum all pairs, and include the term when $i = Q/2$, we obtain

$$H_o(\omega) = e^{-jQ\omega T_s/2}[C(\omega) + jS(\omega)] , \qquad \text{(A3-9)}$$

where

$$C(\omega) = 0.5 + \sum_{i=1}^{Q/2-1} \cos(i\omega T_s) \qquad \text{(A3-10)}$$

and

$$S(\omega) = -2 \sum_{i=1}^{Q/2-1} g'(Q/2 - i)\sin(i\omega T_s) . \qquad \text{(A3-11)}$$

Returning to Eq. (A3-4) and factoring out $z^{-(Q+1)/2}$, we see that

$$H_{CD}(z) = z^{-(Q+1)/2}[z^{-(Q-1)/2} + (z^{1/2} - z^{-1/2})z^{Q/2}H_o(z)] . \qquad \text{(A3-12)}$$

Again letting

$$z = \exp(j\omega T_s) , \qquad \text{(A3-13)}$$

the above expression becomes

$$H_{CD}(\omega) = e^{-j(Q+1)\omega T_s/2}\{e^{-j(Q-1)\omega T_s/2} + 2j\sin(\omega T_s/2)[C(\omega) + jS(\omega)]\} . \qquad \text{(A3-14)}$$

If we apply Euler's formula to $e^{-j(Q-1)\omega T_s/2}$ and expand Eq. (A3-14), we find that

$$H_{CD}(\omega) = e^{-j(Q+1)\omega T_s/2}\{[\cos(\omega T_s(Q-1)/2)$$

$$- 2\sin(\omega T_s/2)S(\omega)$$

$$+ j[-\sin(\omega T_s(Q-1)/2) \tag{A3-15}$$

$$+ 2\sin(\omega T_s/2)C(\omega)]\} .$$

Now we combine the following equations

$$\omega = 2\pi f , \tag{A3-16}$$

$$f_s = 1/T_s = Rf_N \tag{A3-17}$$

and

$$f_N = 2f_c , \tag{A3-18}$$

to see that

$$\omega T_s = \pi f/Rf_c . \tag{A3-19}$$

By substituting Eq. (A3-19) and the formulae for $C(\omega)$ and $S(\omega)$, given in Eqs. (A3-10) and (A3-11), respectively, into Eq. (A3-15), we arrive at the final form of $H_{CD}(f)$, i.e.,

$$H_{CD}(f) = \exp(-j(Q+1)\pi f/2Rf_c)\{\{\cos(\pi(Q-1)f/2Rf_c)$$

$$+ 4\sin(\pi f/2Rf_c)[\sum_{i=1}^{Q/2-1} g'(Q/2-i)\sin(i\pi f/Rf_c)]\}$$

$$+ j\{-\sin(\pi(Q-1)f/2Rf_c) + 2\sin(\pi f/2Rf_c)$$

$$\cdot \left[0.5 + \sum_{i=1}^{Q/2-1} \cos(i\pi f/Rf_c)\right]\}\Big) \; . \tag{A3-20}$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX 4

THE EFFECT OF TIME DELAY ON THE SNR OF THE BASIC PCM TO ADM CONVERTER

Although we have shown that the time delay, γ_o, required to maximize the optimum SNR of the basic PCM to ADM converter is zero, it is interesting to study the relationship between Q_o and the time delay, γ_o. We shall explore this dependence for the case of a white, bandlimited input power spectral density, that is,

$$\begin{aligned} G_x(f) &= T_N \ , \qquad |f| < f_m \ , \\ &= 0 \ , \qquad |f| > f_m \ , \end{aligned} \tag{A4-1}$$

where

$$T_N = 1/2f_m \ . \tag{A4-2}$$

This input power spectrum is one of the examples applied to the SNR statistical analysis.

Since the input power is set to unity, as seen from Eq. (A4-1), we shall begin from the expression for Q_o given in Eq. (4.3-16), with $P_x = 1$, i.e.,

$$Q_o = \frac{1}{1 - \mu^2(\gamma_o)/P_w} \ . \tag{A4-3}$$

The in-band signal power is calculated from

$$P_w = R_w(0) - P_{w_o} \ , \tag{A4-4}$$

where

$R_w(0)$ = the total power in $w(t)$, which is evaluated from Eq. (4.3-38) for this particular example,

and

P_{w_o} = the out-of-band power in $w(t)$, found from Eq. (4.3-56).

For the basic PCM to ADM converter, and this example, these calculations are easily made since Eq. (4.3-56) becomes a short, finite sum.

We must evaluate $\mu(\gamma_o)$ to see the effect of γ_o on the SNR. Starting with Eqs. (4.3-17) and (4.3-45), and the even property of $G_{xw}(f)$, we can express $\mu(\gamma_o)$ by the following integral,

$$\mu(\gamma_o) = \lim_{\varepsilon \to 0} (4/T_N) \int_{\varepsilon}^{f_m} \left[\frac{\cos(2\pi f \gamma_o)}{(2\pi f)^2} - \frac{\cos(2\pi f(\gamma_o + T_N)) + \cos(2\pi f(\gamma_o - T_N))}{2(2\pi f)^2} \right] df \ , \tag{A4-5}$$

where we have employed some trigonometric identities and used the limit as $\varepsilon \to 0$ because the integrand blows up at $f = 0$. After changing variables and using a well-known relationship from a table of integrals,

$$\int \frac{\cos x}{x^n} dx = -\frac{\cos x}{(n-1)x^{n-1}} - \frac{1}{(n-1)} \int \frac{\sin x}{x^{n-1}} dx \ , \ n > 1 \ , \tag{A4-6}$$

we can, with the aid of L'Hospital's rule, calculate all

terms in Eq. (A4-5).

The final expression for $\mu(\gamma_o)$ takes the following form:

$$\begin{aligned}\mu(\gamma_o) = & -(4/\pi^2)\cos(\pi\gamma_o/T_N) \\ & - (2/\pi)(\gamma_o/T_N)\,\mathrm{Si}(\pi\gamma_o/T_N) \\ & + (1/\pi)(1 + \gamma_o/T_N)\,\mathrm{Si}(\pi(1 + \gamma_o/T_N)) \\ & + (1/\pi)(1 - \gamma_o/T_N)\,\mathrm{Si}(\pi(1 - \gamma_o/T_N))\end{aligned} \tag{A4-7}$$

where

$$\mathrm{Si}(\alpha) \equiv \int_0^{\alpha} \frac{\sin x}{x}\,dx \tag{A4-8}$$

and the sine integral, $\mathrm{Si}(\alpha)$, is a common tabulated function. Now we can evaluate Q_o from Eq. (A4-3) for various values of γ_o/T_N. In Fig. A4-1, we show a plot of Q_o versus γ_o/T_N for $|\gamma_o| \leq T_N$. When $|\gamma_o| > T_N$, Q_o will oscillate between 0 and 1.5 dB. We observe that the peak occurs at $\gamma_o = 0$ as expected and after $\gamma_o = \pm 0.2T_N$ the Q_o curve falls off very rapidly. This indicates that we can have a delay of several ADM periods before we suffer a drastic reduction in SNR, but no more than one-fifth of the Nyquist period.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

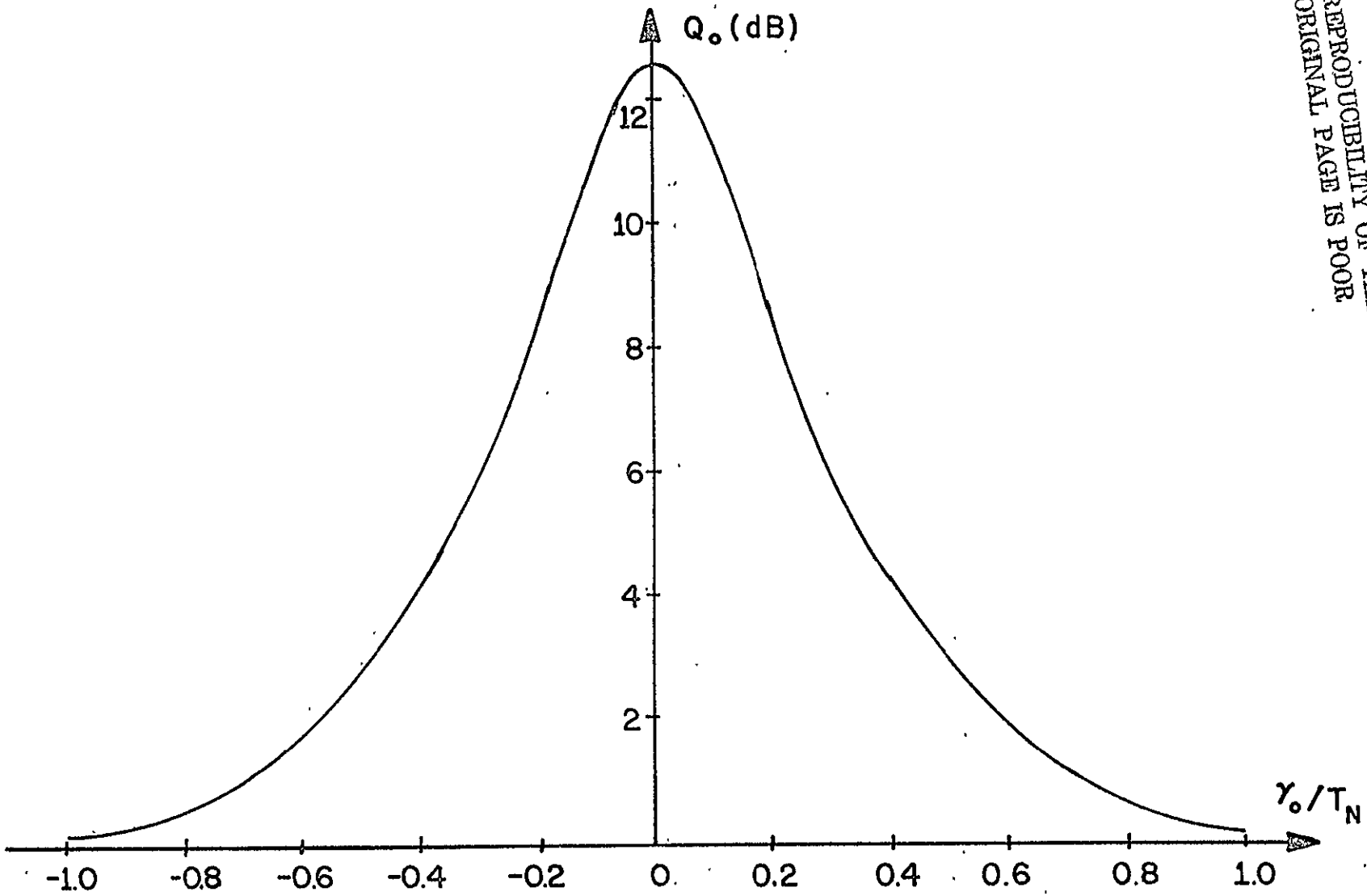


Fig. A4-1. Optimum SNR of the Basic PCM to ADM Converter as a Function of Optimizing Time Delay

REFERENCES

[1] F. de Jager, "Delta modulation, a method of PCM transmission using the 1-unit code," Philips Res. Rept., no. 7, pp. 442-466, Nov. 1952.

[2] H. Van de Weg, "Quantization noise of a single integration delta modulation system with an n-digit code," Philips Res. Rept., no. 8, pp. 367-385, Oct. 1953.

[3] M. R. Winkler, "High information delta modulation," 1963 IEEE International Conv. Record, pt. 8, pp. 260-265.

[4] J. A. Greefkes and F. de Jager, "Continuous delta modulation," Philips Res. Rept., no. 23, pp. 233-246, 1968.

[5] J. B. O'Neal, "Delta modulation quantizing noise analytical and computer simulation results for Gaussian and television input signals," Bell Sys. Tech. J., vol. 45, pp. 117-141, Jan. 1966.

[6] J. E. Abate, "Linear and adaptive delta modulation," Proc. IEEE, vol. 55, pp. 293-308, Mar. 1967.

[7] J. A. Greefkes, "A digitally companded delta modem for speech transmission," IEEE International Conf. Commun., pp. 7-33 - 7-48, June 1970.

[8] N. S. Jayant, "Adaptive delta modulation with one-bit memory," Bell Sys. Tech. J., vol. 49, pp. 321-342, Mar. 1970.

[9] C. L. Song, J. Garodnick and D. L. Schilling, "A variable-step-size robust delta modulator," IEEE Trans. Com-

mun. Tech., vol. COM-19, pp. 1033-1044, Dec. 1971.

[10] L. H. Zetterberg, "A comparison between delta and pulse code modulation," Ericsson Technics, vol. 11, pp. 95-154, 1955.

[11] P. P. Wang, "Idle channel noise of delta modulation," IEEE Trans. Commun. Tech., vol. COM-16, pp. 737-742, Oct. 1968.

[12] D. J. Goodman, "Delta modulation granular quantizing noise," Bell Sys. Tech. J., vol. 48, pp. 1197-1218, May-June 1969.

[13] H. R. Schindler, "Linear, nonlinear and adaptive delta modulation," IEEE Trans. Commun., vol. COM-22, pp. 1807-1822, Nov. 1974.

[14] Song, Garodnick and Schilling, op. cit., pp. 1033-1044.

[15] Ibid., pp. 1033-1039.

[16] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, N.Y.: John Wiley & Sons, Inc., 1967, pp. 674-676.

[17] Transmission Systems for Communications, Members of the Technical Staff, Bell Telephone Laboratories, Winston-Salem, North Carolina: Western Electric Co., Inc., Tech. Pub., 1964, pp. 624-629.

[18] P. J. May, C. J. Zarcone and K. Ozone, "Voice band data modem performance over companded delta modulation channels," IEEE Int. Conf. on Commun., vol. III, pp. 40-16 - 40-21, June 1975.

[19] D. J. Goodman, "The application of delta modulation to analog-to-digital encoding," Bell Sys. Tech. J., vol. 48, pp. 321-343, Feb. 1969.

[20] L. D. J. Eggermont, "A single-channel PCM coder with companded DM and bandwidth-restricting filter," IEEE Int. Conf. on Commun., vol. III, pp. 40-2 - 40-6, June 1975.

[21] Goodman, op. cit., pp. 335-338.

[22] H. Taub and D. L. Schilling, Principles of Communication Systems, N.Y.: McGraw-Hill, Inc., 1971, pp. 172-174.

[23] Eggermont, op. cit., pp. 40-2 - 40-6.

[24] Goodman, op. cit., pp. 321-343.

[25] Eggermont, op. cit., p. 40-6.

[26] B. Gold and C. M. Radar, Digital Processing of Signals, N.Y.: McGraw-Hill, Inc., 1969, pp. 51-70.

[27] T. Ohno, H. Kuwahara, M. Miyata and K. Imai, "Voiceband analog-PCM conversion system using delta modulation," IEEE Int. Conf. on Commun., vol. II, pp. 31-22 - 31-25, June 1976.

[28] J. H. Miller, "Digital-to-digital conversion between an adaptive delta modulation format and companded PCM," IEEE Int. Conf. on Commun., vol. III, pp. 43-29 - 43-34, June 1976.

[29] L. B. Jackson, J. F. Kaiser and H. S. McDonald, "An approach to the implementation of digital filters," IEEE Trans. Audio Electroacust., vol. AU-16, pp. 179-180, Sept. 1968.

[30] A. Papoulis, Probability, Random Variables, and Stochastic Processes, N.Y.: McGraw-Hill, Inc., 1965, pp. 338-339.

[31] Ibid., pp. 337-338.

[32] A. Papoulis, The Fourier Integral and its Applications, N.Y.: McGraw-Hill, Inc., 1962, pp. 53-55.

[33] Papoulis, Probability, Random Variables, and Stochastic Processes, op. cit., pp. 390-394.

[34] Wozencraft and Jacobs, op. cit., pp. 678-683.

[35] Ibid, pp. 348-351.

[36] D. J. Goodman and J. L. Flanagan, "Direct digital conversion between linear and adaptive delta modulation formats," IEEE Int. Conf. on Commun., pp. 1-33 - 1-36, June 1971.

[37] B. C. Kuo, Analysis and Synthesis of Sampled-Data Control Systems, Englewood Cliffs, N.J.: Prentice-Hall, 1963, pp. 82-86.

[38] Ohno, Kuwahara, Miyata and Imai, op. cit., pp. 31-22 - 31-25.

[39] Taub and Schilling, op. cit., pp. 172-174.