

https://dspace.org/

A basis for digital instrumentation to measure quantisation distortion in speech pulse code

Goedhals, Anton Beau

2015-01-21
http://hdl.handle.net/10539/16674

Downloaded from a DSpace repository. https://dspace.org/

A BASIS FOR DIGITAL INSTRUMENTATION TO MEASURE QUANTISATION DISTORTION IN SPEECH PULSE CODE MODULATION SYSTEMS

Anton Beau Goedhals

A Dissertation submitted to the Faculty of Engineering,
University of the Witwatersrand, Johannesburg
in Partial Fulfilment of the requirements
for the degree of Master of Science in Engineering

Johannesburg
April 1979

DECLARATION

I hereby declare that this dissertation is my own work and that it has not previously been submitted for a degree at any university.

A.E. Goedhals

ABSTRACT

Pulse code modulation systems are increasingly important in speech communication. In such systems it may be expected that quantisation distortion will be the predominant cause of signal impairment.

The design of equipment to test and measure quantisation distortion has, until now, been based on an accumulation of successive ad hoc improvements. In the work reported here, it has been sought to establish a formal and explicit structure of the requirements for meaningful and precise quantisation distortion measurement.

The class of test systems in which signal and noise are separated in the time domain has been analysed. The use of narrow bandwidth stimuli for these systems has been proposed. Frequency domain separation systems have also been analysed. A test system of this type has been proposed, and it is based largely upon digital technology.

Simulation tools have been created to investigate aspects of quantisation distortion measurement systems. By-products of this have been a novel FORTRAN/assembly language interface for Data General mini-computers, and a program which designs digital filters. These may be of general interest. An initial use of the simulators has been to quantify the importance of several test stimulus parameters. This has established the feasibility and advantages of the proposed test set, and has determined the design parameter values that may be expected for such systems.

An important parameter of the test stimulus is the amplitude probability density function. The use of gamma distributed rather than accurately Gaussian stimuli has been proposed here. The difficulties of generating such stimuli have been illustrated. A technique has been proposed whereby convenient (Gaussian) stimuli may be used for measurements; the performance for more speech-like signals may then be inferred by a transformation procedure, without explicit synthesis of these signals. This has allowed the calculation of the error incurred by the conventional assumption that speech is approximately normally distributed. A re-evaluation of usual loading levels for codecs has been suggested. It has been established that sophisticated techniques for generating pseudo-random sequences with arbitrarily prescribable distribution are of little importance in quantisation distortion measurement.

Embryo facilities have been established to allow laboratory tests of pulse code modulation systems. Encoder and decoder hardware for a

typical system have been built and interfaced to a minicomputer. Software models of the encoder and decoder have been incorporated in the simulators. The quantisation noise performance of an encoder may be measured, and the additional impairment due to hardware imperfections may also be measured.

Pseudo-random signals with a wide variety of distributions have been filtered using various digital filters. In all non-trivial cases, the filtered signal has been found to have a distribution which appears approximately Gaussian. It has been possible to demonstrate quite rigorously that this phenomenon cannot be accounted for by the central limit theorems. This has posed a fundamental question as to the theoretical basis for quantisation distortion measurement in particular.

It is expected that many of the ideas of quantisation distortion measurement will find wider application in performance measurement of completely analog systems.

To

Mrs Walsh

and

Miss B Thompson

ACKNOWLEDGEMENTS

The work described here was done in the Department of Electrical Engineering at the University of the Witwatersrand, Johannesburg. The initial two years of this work were financially supported jointly by the University and the Council for Scientific and Industrial Research. This support directly created the opportunity to pursue this work, and I am grateful for it.

This project was initiated on the basis of work being done by Professor H.E. Hanrahan and Mr A.J. Walker, and I thank them for their guidance. Perhaps more important than his supervision of this project, Professor Hanrahan has introduced me to a particular engineering style.

Anne Cuthbert has helped me in the course of this work, and particularly in producing the typed versions of this dissertation. I appreciate also the less direct contributions of many people, and I thank them.

CONTENTS PAGE

ABSTRACT ... iii
ACKNOWLEDGEMENTS ... vi
LIST OF ILLUSTRATIONS ... xii
LIST OF ABBREVIATIONS ... xiv

1 *INTRODUCTION*

1.1 Background ... 1
1.2 The Aims of the Present Work ... 2
1.3 Review of Approaches to Quantisation Distortion Measurement ... 2
1.3.1 Time Domain Subtraction ... 3
1.3.2 Frequency Domain Subtraction ... 7
1.3.3 A Commercially Available Quantisation Distortion Measuring Set - The Marconi TF2807 ... 8
1.3.3.1 The Noise Source ... 8
1.3.3.2 The Filters ... 9
1.3.3.3 The Attenuator ... 10
1.3.3.4 The RMS Meter ... 10
1.4 Summary of Established Requirements for a Frequency Domain Subtraction System ... 10
1.4.1 The Type of Test Signal ... 10
1.4.2 Parameters of the Test Signal ... 12
1.4.3 Parameters of the Test Set ... 12
1.5 Outline of the Present Work ... 13

2 SYSTEM DEFINITION PHILOSOPHY

2.1 Specification of the Primary Goal - A Range and Accuracy Requirement ... 15
2.2 An Evaluation of Time Domain Subtraction ...
2.3 An Evaluation of Established Requirements for *Frequency Domain Systems* ... 24

CONTENTS

PAGE

2.3.1 Frequency Spectral Requirements 24
2.3.2 Statistical Properties Required 28

2.4 Measurement Techniques with Frequency Domain Subtraction 35

2.4.1 An Evaluation of the Established Measurement Technique 35
2.4.2 A Proposed Measurement Technique 36

2.5 Outline of the Proposed Test System 42

2.5.1 The Pseudo-Random Noise Source 42
2.5.2 Filters 44
2.5.3 Attenuators and Data Converters. .. . 45

2.6 A Test System Simulator 46

2.6.1 The Purpose of Simulation 46
2.6.2 Hardware Considerations 47

2.7 A Summary of the Ideas Defining the Simulator . 49

3 TEST SYSTEM SIMULATION

3.1 Software Description 51

3.1.1 The Two Simulation Packages 51
3.1.2 Software Modules in the Nova Package .. 57

3.1.2.1 A Flexible Program to Aid Specification of Arbitrary PDF's: PROBSPEC 59
3.1.2.2 A Program to Configure the Noise Source to Produce a Specified Distribution: ALICUT 61
3.1.2.3 A Program to Generate Pseudo-Random Sequences with Prescribed Distribution : RNGEN 62
3.1.2.4 The Digital Filters 65
3.1.2.5 Programs to Simulate a Linear Digital to Analog Converter (QUANTO), a Companding PCM Encoder (IDLPCM) and a Companding D[illegible]CONVERT) 70

CONTENTS

PAGE

3.2 Software - Hardware Interaction 72

3.2.1 The Fortran/Hardware Interface 72

3.2.2 Assembly Language Programs 75

3.3 A Summary Description of Test System Software 76

4 TEST SYSTEM SIMULATIONS PERFORMED

4.1 Probability Distribution Modelling 78

4.2 The COMDAC SQNR versus Test Stimulus Level Characteristic - Measurement of a Reference Curve 80

4.3 Variation of Measured SQNR as a Function of Test Length 80

4.4 Measured SQNR as a Function of Test Stimulus Frequency Domain Parameters 83

4.4.1 Measured SQNR Variation with Stimulus Centre Frequency 83

4.4.2 Measured SQNR Variation with Stimulus Bandwidth 83

4.5 Determination of Data Converter Signal Level to Optimise Test Set Performance 87

4.6 Investigation of Test Set Design Trade-Offs between Data Converter Resolutions and Filter Orders 88

4.6.1 Effects of Constrained Data Converter Resolutions 90

4.6.2 Filter Requirements with Practicable Data Converter Resolutions 91

4.7 Summarising Comments on the Simulations Performed. ... 94

5 FURTHER PROPOSALS AND CONCLUSIONS

5.1 The Response of Linear Filters to Stimuli which have Non-Gaussian Probability Distribution 95

CONTENTS PAGE

5.1.1 General Analytic Techniques for Systems with Non-Gaussian Stimuli 95

5.1.2 The Applicability of the Central Limit Theorems 96

5.1.3 The Applicability of the Liapounoff Theorem 99

5.1.4 A Necessary Condition for the Central Limit Theorems 103

5.2 A Proposed Test Algorithm 104

5.2.1 Theoretical Basis .. .., ... 104

5.2.2 Numerical Significance ... .., ... 112

5.3 A Proposal for a Deterministic Test Stimulus 114

5.3.1 Approach and Motivation 114

5.3.2 Results of Simulations to Gauge Feasibility 116

5.4 Preliminary Comments on Hardware Test Sets 118

5.5 Noise Sources for Future Work 120

5.6 The Test System Sampling Frequency 121

5.7 Concluding Comments .., ... 123

BIBLIOGRAPHY CITED 125

CONTENTS PAGE

APPENDICES 130

CONTENTS OF APPENDICES 130

APPENDIX A : A Laboratory PCM System 133

APPENDIX B : A 12-Bit Digital to Analog Converter for the Nova Computer 155

APPENDIX C : A Revision of the Dual 10-Bit Digital to Analog Converter Facility on the Nova Computer 157

APPENDIX D : Simulator and Hardware Testing Software for the Nova Computer 160

APPENDIX E : Simulation and Utility Software for the IBM 370 Computer 180

APPENDIX F : The Combined Transformations used for Digital Filter Design 194

APPENDIX G : Numerical Data and Results from Simulations . 197

APPENDIX H : SQNR Curve Transformation Program : XFMG2S .. 209

APPENDIX I : A Simulation Program for Deterministic Stimuli: SINES 212

LIST OF ILLUSTRATIONS

Figure		Page
Fig. 1	Conceptual Outline of a Time Domain Subtraction System.	3
Fig. 2	An Approach to Realising a Time Domain Subtraction System.	4
Fig. 3	Block Diagram of an Implemented Time Domain Subtraction System.	5
Fig. 4	Block Diagram of the Marconi TF2807.	9
Fig. 5(a)	A Functional Representation of a General PCM System.	19
Fig. 5(b)	Development of a Simpler Model of a PCM System.	19
Fig. 5(c)	Analytic Representation of a Time Domain Subtraction System.	20
Fig. 6	Offset between Compressor and Expander.	26
Fig. 7	Long Term Average PDF's of Speech, and Analytic Distributions (after Richards, 1973, p65).	30
Fig. 8	Gaussian PDF Approximation Achieved by Rolls and Webster (after Rolls and Webster, 1974, p94).	32
Fig. 9	Possible PDF's in the Test Stimulus Generator.	34
Fig. 10	The *Single Band Injection Measurement Technique*.	35
Fig. 11	The Dual Band Injection Technique Proposed.	37
Fig. 12	Error Spectra for a Simulated DPCM System (after Mc Donald, 1966, p1138).	41
Fig. 13	Functional Blocks Comprising an Experimental Test System.	43
Fig. 14(a)	The Basic Element of Simulation Package Structure on the Nova Computer.	53
Fig. 14(b)	Disk File Nodes and Program Link Structure.	53
Fig. 15	The Real-Time Program Link for Hardware Measurements.	54
Fig. 16	Elements of Simulation Package Structure on the IBM 370.	56
Fig. 17	Operation of the Simulation Package on the IBM 370.	57
Fig. 18	Nova Software Skeleton.	58
Fig. 19	Approaches to Distribution Generation.	60
Fig. 20	Design and Implementation of Bandpass Digital Filters.	66
Fig. 21	Filter Coefficient Calculation in Program BUTTER.	68
Fig. 22	Transform Procedure using Pole Pairs in Program BUTTER.	69
Fig. 23	Transfer of a Data Block from FORTRAN to Assembly Driver.	73
Fig. 24	Variety of Random Source Target Distribution Shapes Used.	81
Fig. 25	The SQNR Versus Test Stimulus Level Characteristic of the COMDAC.	81

Figure		Page
Fig. 26	Variation of Measured SQNR with Test Sequence Period.	84
Fig. 27	Variation of SQNR Measured as a Function of Noise Band Centre Frequency.	85
Fig. 28	Measured SQNR as a Function of Stimulus Bandwidth.	86
Fig. 29	Test Set Data Converter and Attenuator Sections.	87
Fig. 30	Test Set SNR as a Function of Digital Attenuation.	89
Fig. 31	Differences in SQNR Measured with Stimulus PDF's Tending to a Gamma Distribution, Relative to Gaussian Stimulus Measurements.	113
Fig. 32	Outline of Stimulus Source Section of Simulation Package SINES for IBM 370.	116
Fig. 33	Amplitude Distribution of Narrowband Stimulus Synthesised from Sinusoids.	117
Fig. 34	Speech-Like Distribution of Stimulus Synthesised from Modulated Sinusoids.	117
Fig. 35	Source with Stimulus Stored in ROM.	118
Fig. 36	Source with Off-line Calculation of Stimulus Sequence.	118
Fig. 37	Deterministic Stimulus Source.	119
Fig. A1	Functional Blocks in the PCM System Hardware.	134
Fig. A2	PCM System Analog Circuitry (Channel A).	146
Fig. A3	Analog Switch Control and TTL-CMOS Interface.	147
Fig. A4	COMDAC Data Selector and Encode/Decode Logic.	148
Fig. A5	Mode Control Logic and (Channel A) Data Input.	149
Fig. A6	A Conceptual Representation of the PCM System Timing.	150
Fig. A7	Timing and Control Logic for Input Sampling; Channel B Data Input.	151
Fig. A8	Timing and Control Logic for Output Sampling; Forced Lock Conversion Clock.	152
Fig. A9	Parallel Data Output; Sequence Flags.	153
Fig. A10	The Busy and Done Flags.	154
Fig. A11	Details of Data Selector Using NAND Gates.	154
Fig. A12	Timing Waveforms at Start of Encode Sequence.	154
Fig. B1	The 12-Bit Digital to Analog Converter Circuitry and Associated Timing and Control Circuits.	156
Fig. C1	Buffered Latches and Digital to Analog Circuitry.	158
Fig. C2	Single Bit Output Channel; Timing, Flags and Control Circuitry.	159
Fig. G1	PDF's for a Gaussian RQPROB.	199
Fig. G2	PDF's for a Rectangular RQPROB.	199
Fig. G3	PDF's for RQPROB an Arbitrary Spike.	200
Fig. G4	PDF's for RQPROB Similar to the Distribution of a Sinusoid.	200

Figure		Page
Fig. G5	PDF's for RQPROB a Twin Triangular Distribution	201
Fig. G6	PDF's for RQPROB a Twin Triangular Distribution, and Send Filter Bandwidth 1-3900 Hz.	201

LIST OF ABBREVIATIONS

AC	alternating current
A/D	analog to digital
A-law	family of companding laws, dominant in European systems
ASCII	American Standard Code for Information Interchange
bit	binary digit
CCITT	Comite Consultatif International Telegraphique et Telephonique
CEPT	European Committee for Posts and Telegraphs
CMOS	complementary metal oxide semiconductor
codec	encoder decoder combination
COMDAC	companding D/A converter, registered trade mark of PMI
ΔΣ	delta sigma (modulation)
D/A	digital to analog
dB	decibel
dBm	decibels referred to one milliwatt
dBm0	decibels referred to an arbitrarily defined zero level
DC	direct current
DDOS	Decision disk operating system
DEMOD	demodulator
DGC	Data General Corporation
DM	delta modulation
DPCM	differential pulse code modulation
EPROM	erasable and programmable read only memory
FORTRAN	formula translation language, a programming language
FSR	feedback shift register
H-P	Hewlett Packard
Hz	hertz
IBM	International Business Machines
kHz	kilohertz
LVL	level

MOD	modulator
MODEM	modulator and demodulator combination
ms	millisecond
μ-law	a specific family of companding characteristics
μP	microprocessor
OS	a specific operating system on the IBM370
PCM	pulse code modulation
PDF	(amplitude) probability density function
PMI	Precision Monolithics Incorporated
PROM	programmable read only memory
QD	quantisation distortion
RAM	random access memory, read/write memory
RC	resistance capacitance
RHS	right hand side
RLC	resistance inductance capacitance
RMS	root mean square
ROM	read only memory
SAR	successive approximation register
SNR	signal to noise ratio
SQNR	signal to quantising noise ratio
THD	total harmonic distortion
TTL	transistor-transistor logic
μ-law	family of companding laws, used in North American systems
μP	microprocessor
WITS	Witwatersrand Interactive Terminal System
WOM	write only memory, (some floppy diskettes)

1 INTRODUCTION

1.1 Background

There is a growing preference for communications systems based on discrete representations of information. There have arisen increasingly compelling reasons to convey speech signals in fully discrete form, particularly in switched networks. Pulse code modulation (PCM) may be seen as the most important waveform coding technique for speech. Certainly, PCM is for the present purposes an adequate conceptual representation of all the non-vocoder techniques.

The major source of signal impairment in a well-designed PCM system is known to be amplitude quantization. It is necessary to measure quantisation distortion for two purposes. At the design stage, the quantisation distortion performance of encoders should be included in an assessment of their relative merits. In the maintenance of codecs in service, quantisation distortion tests are a ready verification of the end-to-end performance of a link. The existence of a fault at virtually any stage in the link will be reflected in the quantisation test results.

It is now attractive in some applications also to transmit signals in their digital form, and the PCM links to be tested have frequently come to include very high fidelity tape recorders, (Cohen, 1977).

A philosophy of meaningful quantisation distortion measurement has been developing. In general, such measurements rely upon the injection of a test signal into the PCM system. Pseudo-random test signals have been shown to be desirable. The amplitude probability density function has been identified as an important parameter of the test signal.

A few commercial instruments have become available for measuring quantisation distortion. These testers are based on relatively simple pseudo-random sources. They achieve amplitude probability densities which roughly approximate to the desired Gaussian distribution.

The starting point of the work to be described was the recent development by Walker of a new technique for generating pseudo-random sequences. This technique allows that the probability distribution of the generated sequence be arbitrarily prescribed, (Walker, 1974, 1977). The technique has promised to be useful in generating test signals which attain better approximations to the desired probability distribution.

1.2 The Aims of the Present Work

The aim of the work to be described has been to investigate the usefulness of Walker's random source techniques in the generation of test signals for measuring quantisation distortion in PCM systems. This investigation has sought to answer several broad questions.

(a) Can Walker's technique be used to generate a test signal which approximates more closely to the desired probability distribution than does a signal generated by conventional feedback shift register techniques?

(b) How are any improvements in the test signal probability distribution reflected in terms of significance of the distortion measurement?

(c) What is the feasibility of engineering a test system using a more sophisticated pseudo-random source, and how is such a design best approached?

The problem has thus been to design for evaluation purposes a test system incorporating a flexible pseudo-random source. The level of detail and of implementation of such a design has had to be sufficient so that a fair comparison could be made with conventional designs.

1.3 Review of Approaches to Quantisation Distortion Measurement

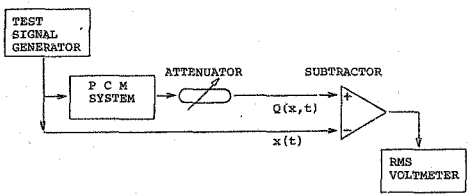
The purpose of this section is to define those test systems which will be regarded as conventional in later comparisons with the system to be proposed. It is sought to outline conventional systems in sufficient detail to provide a basis for motivation of the system proposed in Chapter 2. The circuit functions outlined also provide a basis for the evaluation of test signal requirements in Section 1.4.

It is felt that all methods of quantisation distortion (QD) measurement must necessarily involve the same basic elements. A signal is injected at the input to the PCM system. This is a fundamental requirement because although quantisation noise may be regarded as an additive noise source, this source exists only when there is a signal in the channel. At the output of the PCM system, the input test signal is 'subtracted' from the system output signal. The remaining component of the output signal is the 'noise' - primarily quantisation noise in a PCM system.

It is possible to conceive of at least two classes of QD measuring systems, distinguished by the way in which the test input signal is subtracted from the PCM system output. The conceptual subtraction may be done in the time domain, or in the frequency domain. These are not equivalent, as the detailed considerations of Sections 2.2 and 2.3 demonstrate.

1.3.1 Time Domain Subtraction

The time domain approach appears more direct in principle, and is shown in Fig. 1.



x (t) = test signal

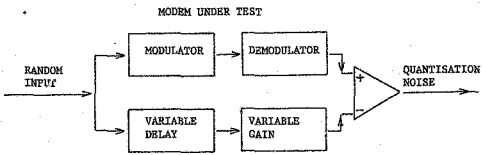
Q (x,t) = version of test signal reconstructed from quantised x(t)

Figure 1 Conceptual Outline of a Time Domain Subtraction System.

The attenuator is usually required, and is to allow the overall gain of the PCM system signal path to be set to unity.

The general problem of QD measurement is to measure the signal to noise ratio (SNR) rather than the noise power. The information required for practical systems is a plot of SNR as a function of input signal level, over a wide range of signal levels. Typically, the same meter system is used to measure both the noise power and the signal power. (The switching of the meter input has not been shown in Fig. 1.) A true *RMS meter is required because* the quantising error waveform is not well defined ab initio.

A system using time domain subtraction has been described by Kikkert (1971). Kikkert has made the valuable point that the system he has described may be used for measurements on *either PCM systems or on* delta modulation (DM) systems. This point is considered in later discussion. Kikkert has given a visualisation of the requirements as shown in Fig. 2.



MODEM = MOdulator and DEModulator combination.

Figure 2 An Approach to Realising a Time Domain Subtraction System.

A major problem in implementing a time domain subtraction system is the provision of the variable delay. The variable delay must clearly be set to match the signal delay through the modem under test. Kikkert, noticeably an investigator of DM, has described an ingenious approach using delta sigma modulation. The scheme is shown in Fig. 3, and has the noteworthy attribute that the quantisation noise introduced in the process of generating the test signal may be cancelled by that introduced into the delayed test signal.

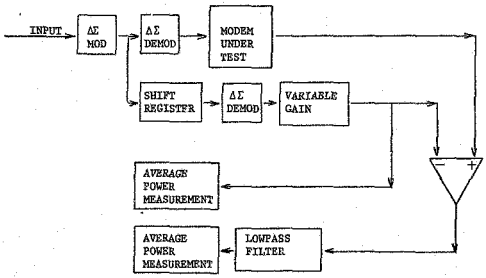


Figure 3 Block Diagram of an Implemented Time Domain Subtraction System.

Kikkert has suggested that for single integration DM systems, it is sufficient to provide a delay equal to one sampling period. The general validity of this assumption for PCM systems is discussed in Section 2.2. However, delays which are integral multiples of the $\Delta\Sigma$ modem sampling period could be obtained by varying the number of stages used in the shift register. Continuous variations of these delays could be obtained, if required, by slightly varying the $\Delta\Sigma$ modem sampling frequency.

It is necessary that the two delta sigma demodulators shown in Fig. 3 be closely matched. Kikkert has constructed a system where the error due to demodulator mismatch is at least 50 dB down on the test signal level. He has suggested that this allows signal to noise ratios up to about 45 dB to be 'accurately measured'.

Kikkert has used voltage rectification and averaging to measure the signal and noise level. This assumes the same form factor for signal and for quantisation noise. The uncertainty in measured signal to noise ratio due to this assumption has not been determined.

Kikkert has raised objections to the testing of DM and PCM systems by means of sinusoidal test signals. He has used computer simulation to compare SNR's obtained from an 800 Hz sinusoid with SNR's obtained with simulated speech. These simulations modelled a single integration DM, but it is noteworthy that the different test signals reportedly gave maximum SNR's differing by 10,5 dB. It may possibly be assumed that the simulation for the sinusoidal input modelled a frequency domain subtraction process, but the discrepancy is nevertheless alarming.

Kikkert has also done a set of tests on a hardware system using an 800 Hz sinusoid as the test signal. He has given two curves of SNR versus input signal level, one obtained from a time domain subtraction technique and the other from a frequency domain subtraction technique. Powers were measured as the square of the average voltage. The results differ by 1,5 dB in the mid-range of the curve, and Kikkert regards this as 'very good agreement'.

A further set of SNR curves has been given by Kikkert. One curve was obtained for tape recorded speech input to the hardware system shown in Fig. 3. The other curve was derived from a computer simulation. It is not known whether the computer simulation used a digitised version of the tape recording or a computer speech model. The results are similar to within 1 dB for moderate input levels.

Kikkert's paper is the only description of a practical time domain subtraction system known to the author. The paper has been of value as an indication of what sort of performance has been achieved by practical time domain subtraction schemes.

In the present work, a decision has been made to investigate frequency domain subtraction rather than time domain subtraction. This decision cannot be regarded as verifiably the best approach, because

sufficient work to decide this has not yet been done in the field of quantisation distortion measurement. The basis for the decision to follow a frequency domain approach is a matter for discussion, as is detailed in Chapter 2.

1.3.2 Frequency Domain Subtraction

The traditional technique of QD measurement by this means is similar to the usual technique for measuring total harmonic distortion of completely analog systems. A narrowband test signal is injected into the PCM system. A narrowband filter is then used to reject the input signal component of the system output, the remaining output component being the 'noise' to be measured.

Various types of test signal have been tried - sinusoid, noise with a Gaussian amplitude probability density distribution, and pseudo-random noise with an approximately Gaussian distribution. The QD measured is dependent on the amplitude probability density distribution of the test signal. This is intuitively reasonable because the amplitude distribution 'weights' the extent to which each quantisation interval in a companding encoder contributes to the quantising error. Cattermole (1969, pp 125-241) has given a very detailed and useful analysis of quantising noise generation and the role of the probability density function. A brief analysis is given in Section 2.3.2.

Rolls and Webster (1974) have compared the three previously mentioned test signals. It is emphasised that, in the present work, consideration has been given primarily to the performance of QD measurements on PCM/DM systems intended for speech communication only. Similarly, Rolls and Webster have considered tests of speech links only. The point is made that the QD of speech links should be measured by injecting a test signal which is similar to 'average' speech. One similarity required is that the amplitude probability density distributions should be as similar as possible.

Natural noise sources have Gaussian distributions usually, and this is a first approximation to the speech distribution. However, such sources are not always repeatably constructed, and require long integration times to give repeatable measurement results. These problems are largely obviated by the use of pseudo-random and digitally implemented generators.

Rolls and Webster have presented 'a derivation of a set of parameters for an optimised test signal'. This is the basis of the Marconi TF2807 PCM Multiplex Tester which was designed for the British Post Office and which is presently the most common commercial QD measuring apparatus. This instrument will be considered in more detail in the next section, because it represents an existing test system standard against which the performance of the proposed system may be compared.

Rolls and Webster have developed their parameters on an assumption of the principles of quantisation distortion measurement discussed in detail by *Waddington* (1969, 1972). Waddington was constrained to consider only relatively simple random sources, primarily naturally occurring noise sources.

Rolls and Webster have shown the substantial advantages of using a pseudo-random source. They, in turn, have been constrained to consider only simple pseudo-random sources, typically a single feedback-shift-register plus analog narrowband filters. This has given approximately Gaussian distributions.

1.3.3 A Commercially Available Quantisation Distortion Measuring Set - The Marconi TF 2807

A functional block diagram of the QD measurement section of the Marconi TF 2807 is shown in Fig. 4.

It is noteworthy that a notch reject filter is not used in measurement of the noise power. Rather, a bandpass filter is used to sample the noise power spectrum.

1.3.3.1 The noise source

This binary source has a broad power density spectrum which falls off with increasing frequency (f) according to $\left\{\frac{\sin f}{f}\right\}^2$. The spectrum may, however, be regarded as flat over the 450 Hz - 550 Hz region utilised.

A 17 stage shift register is used with feedback from stages 3 and 17 and a clock frequency of 370 kHz. This particular feedback shift register (f.s.r.) was selected from a limited set of suitable f.s.r.'s determined by computer simulation. No details of this simulation have been published. The performance of this and many other f.s.r. sources in conjunction with various filters has been 'extensively' tested. Consequently, it is felt that the TF2807's vital statistics represent the practical limit to the evolution of this type of noise source for this application. The

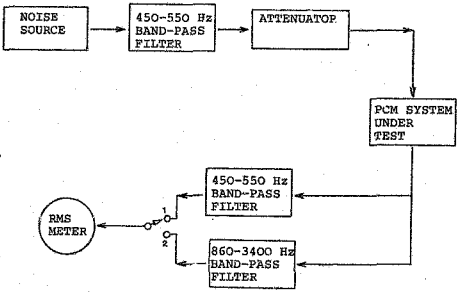


Figure 4 Block Diagram of the Marconi TF2807.

vital statistics are :

sequence time about 350 ms

spectral line spacing 2,8 Hz

peak-to-rms ratio 10,5 dB

Probability distribution (after filtering) is approximately Gaussian. This approximate distribution is shown in Figure 8, where its significance is discussed in Section 2.3.2.

1.3.3.2 The filters

Two narrowband filters (450-550 Hz) are shown Figure 4. These two filters are the same design for convenience. An active RC filter was chosen above a passive RLC filter, because it gave a more Gaussian output. The filters are critical, and all three are implemented as analog filters.

1.3.3.3 The attenuator

This is manually set to adjust the send level over the wide range of -60 dBm0 to +10 dBm0 in 1 dB steps. The 0 dBm0 level is also adjustable in the range -16 dBm to +2 dBm in steps of 0,2 dB.

These figures are given to show that the attenuator is not negligibly inexpensive, and also to give an idea of the precision required.

1.3.3.4 The RMS meter

This is also a significantly demanding analog system to manufacture and keep within calibration. A precision rectifier circuit, an analog voltage squarer, and an integrator and sampler are used. A meter accuracy of 0,5dB has been achieved with a settling time of 2 seconds. QD noise is particularly difficult to measure in the system overload region, where a 20 dB peak-to-RMS ratio of distortion components may be anticipated. The RMS power measurement section has been an important consideration in formulating the test system to be proposed in Section 2.5.

1.4 Summary of Established Requirements for a Frequency Domain Subtraction System

The purpose of this section is to sketch the QD measurement requirements which have been established by previous workers in this field. These conventional requirements have evolved in a practical instrument environment. The intention is to suggest that test signal and system parameters must be more precisely determined, in order to be consistent with the improved accuracy targets of the more complex new designs under examination in the present work.

1.4.1 The Type of Test Signal

Rolls and Webster (1974) have established the desirability of using pseudo-random noise as a test stimulus. They have considered frequency domain subtraction systems, but it is equally desirable for time domain subtraction that the test stimulus be pseudo-random noise.

The noise-like quality is useful in that it gives a smooth QD versus stimulus level response curve. The smoothness of this curve is felt to be advantageous because it allows the curve to be determined by relatively few experimental points. In practice 20 to 30 measurements are taken to establish the curve. A noise-like signal may also have a probability density function which is similar to that of speech.

The attribute of pseudo-randomness is attractive from two points of view. Pseudo-random sources, as opposed to true noise sources, are generally digital in nature. This facilitates the manufacture of sources with easily reproducible properties. The continuing performance of a true noise source is less assured because component ageing effects are more evident in analog systems.

A pseudo-random source is also preferable in that its statistical properties are completely defined within one period of the output sequence. The statistical properties of a segment of such a pseudo-random sequence may also be determined precisely. The output sequence, or any desired segments of it, may be repeatably generated with a certainty as to the 'statistical' properties of the signal. Rolls and Webster have considered various linear-feedback shift register sources, so the stimuli have traditionally been periodic. A true noise source, on the other hand, cannot be precisely characterised for short segments of its output. It cannot be guaranteed that an output of conveniently short duration will have an amplitude distribution sufficiently close to a prescribed distribution. For random sources which are Gaussian, the bandwidth-time law may be used to estimate the power measurement time required.

Consider an estimate $\overline{V_e^2}$ of the mean square amplitude $\overline{V^2}$ of a stationary noise signal. Let the noise have bandwidth B, and let the estimate be an average formed over a measurement time T. Successive measurements will show $\overline{V_e^2}$ to have a normal (Gaussian) distribution with standard deviation of $\sigma(\overline{V_e^2})$. A form of the bandwidth-time law (Hanrahan, 1978) gives

$$\frac{\sigma(\overline{V_e^2})}{\overline{V^2}} = \frac{1}{\sqrt{BT}} \qquad (1.4 - 1)$$

This relation is useful in cases where $BT \geq 5$ (say) (Bendat and Piersol, 1966, p 187). This requirement may be assumed in the present work.

A typical 100 Hz bandwidth gives a 100 second measurement time to reach an uncertainty of 0,1 dB. The measurement may be expected to fall within this limit in 95 percent of cases, (Bendat and Piersol, 1966, p 242). This measurement period is almost intolerably long for routine measurements.

Rolls and Webster have noted that the use of a pseudo-random source allows that any other unpredicted properties of the filtered noise source actually realised be defined. These include the crest factor, and the

effects of non-linear and non-ideal circuit elements. Non-linear elements, particularly those susceptible to saturation, can be expected to extraneously modify a Gaussian distribution.

1.4.2 Parameters of the Test Signal

When a periodic stimulus is used, it is sensible to make the measurement time equal to one signal period. Rolls and Webster have set the period requirement as less than 0,5 seconds.

Some other stimulus parameters have been fixed on the basis of making the stimulus simulate narrow band limited true noise. The probability density function should be Gaussian. The probability of true Gaussian noise being 13 dB above the root mean square value is 0,002 per cent. Hence a crest factor of 10 to 12 dB is acceptable for the test stimulus.

All frequencies within the narrow band of the test stimulus should be equi-probable. This has been approximated by sufficiently closely spaced spectral lines, 2,8 Hz apart in the Marconi instrument. It is noteworthy that the requirement of a flat spectrum within the test stimulus bandwidth has the merit of tending to ensure a flat quantisation noise spectrum, as has been shown by Cattermole (1969, p 239).

Rolls and Webster have suggested that the bandwidth of the test signal 'will have no real significance on the test result providing it is small compared with the system input bandwidth'. Similarly, there is no reference in the available literature as to quantitative attempts to select the most suitable centre frequency for the test stimulus. Cattermole has used a figure of 500 Hz in his studies, as have Rolls and Webster. Various Telecommunication Authorities have used sinusoidal test stimuli with frequencies ranging from 300 Hz to 1100 Hz. Rolls and Webster suggest that 'the frequency should be in the lower end of the input frequency band'.

1.4.3 Parameters of the Test Set

The band edge frequencies required for the noise measurement filter appear also to have been arbitrarily fixed within reasonable limits. One factor has been the provision of a guard band between the (send) filter defining the test stimulus and the noise measurement filter.

A narrowband signal level measuring filter is essential at the receive side only if high QD levels are present. This filter need not have such good characteristics as the send and noise measurement filters. The filter defining the test stimulus before input to the PCM system is referred to

subsequently as the 'send' filter. The filters operating on the PCM system output signal to resolve the stimulus and distortion powers are referred to subsequently as the 'signal' and 'noise' filters.

An important idea raised by Rolls and Webster is that of the SNR indicated by the test set when it is looped back-to-back. They have considered measuring signal to quantising noise ratios up to 36 dB. In order to measure this to within 0,1 dB, the test set back-to-back signal to noise ratio must be 20 dB better, i.e. 56 dB. The idea of 20 dB excess test set SNR in order to measure within 0,1 dB, can easily be shown to apply regardless of the magnitude of signal to quantising noise ratio to be measured.

Frequent reference has been made to the work of Rolls and Webster at Marconi. Their work is the most useful standard for the present purpose because they (and Waddington) have considered carefully the requirements of QD measurement. The Federal German Post Office had, previous to the work at Marconi, presented QD test system proposals to the CCITT. These involved a 10 stage shift register source, 10 kHz clock, 200 Hz wide bandpass filter, and gave a 10 Hz spectral line spacing. Rolls and Webster have given extensive comparative test results showing the Marconi source to give results closer to those obtained with true noise.

1.5 Outline of the Present Work

The present work has been to investigate the usefulness of Walker's flexible pseudo-random source techniques in the generation of test stimuli for QD measurements on PCM systems. This has been resolved to the design of a test system incorporating a flexible pseudo-random source, with the aims indicated previously in Section 1.2.

The available literature on QD measurement has been surveyed. The philosophy underlying present QD measurement techniques and corresponding requirements placed upon elements of present systems have been appraised. The aim has been to examine the requirements explicitly, and to state the conclusions formally as they apply to the test system aimed at higher accuracy than present systems.

A test system which differs from existing designs at the circuit functional block level is proposed in Section 2.5. The proposals have taken cognizance of both Walker's pseudo-random source techniques as well as recent developments in hardware component technology. Much of the present work has been to examine the feasibility of such a system, establish the major design parameters, and evaluate the merits of such an

approach in terms of the significance of the QD measurement.

In order to obtain quantitative results in the above investigations, a flexible simulator computer program has been written to run on an IBM 370/158 computer. This simulator allows a variety of test systems and parameters to be modelled in conjunction with a typical PCM system. Another version of the simulator has been written to run on a Data General Nova 1200 minicomputer. A digital to analog converter has been interfaced to the Nova. This provides a facility for QD measurements using the proposed test system and techniques to be made on hardware systems. A hardware version of the PCM encoder modelled in the IBM 370 simulations has been built and interfaced to the Nova. The facility to perform comparative tests on the simulator and the hardware has been provided.

A test method has been proposed whereby tests are done for both high and low frequency stimuli, and for which fewer assumptions need be made about the quantisation noise spectrum. A facility has been provided to investigate this method by computer simulation.

Two ideas of fundamental importance have been generated in the present work.

A simple algorithm has been proposed which allows a signal to quantising noise ratio curve established using Gaussian stimuli to be transformed to the curve which would have resulted if stimuli with speech-like probability distribution had been used. A procedure to calculate the error due to the use of a Gaussian stimulus as opposed to a statistically speech-like stimulus, has been demonstrated.

The work has also entered the area of the theoretical basis for QD measurement by non-Gaussian stimuli. A distribution limiting effect has been observed when filtering non-Gaussian signals. It has been shown quite rigorously that this effect cannot be anticipated in terms of the central limit theorems. This result is surprising in that it is contrary to commonly accepted views.

2 SYSTEM DEFINITION PHILOSOPHY

The purpose of this chapter is to examine more closely particular aspects of QD measurement. The intention is to discuss explicitly those aspects which were the foremost considerations in the definition of the general system requirements. Herein lies the motivation for the form of the system proposed in outline in Section 2.5 and detailed in simulation in Chapter 3.

It is convenient to abbreviate the term signal to quantising noise ratio to SQNR, and this distinguishes it from the unqualified signal to noise ratio (SNR).

2.1 Specification of the Primary Goal - a Range and Accuracy Requirement

The essential aim has been to define a system which is capable of measuring signal to quantising noise ratios up to 43dB and meaningful to within (say) 0,25 dB. An SQNR measurement is meaningful only insofar as it relates to subjective assessments of speech link quality; this point is discussed in Section 2.3. This specification was not fixed for wholly arbitrary reasons, but was decided by the following considerations.

It is felt that a need exists for measurement of SQNR's of up to some 65 dB. This need arises in conjunction with speech-like signals in audio equipment employing PCM. Such equipment is primarily studio equipment at present but a variety of PCM audio adapters are appearing for consumer video recorders (Cohen, 1977).

Experimental adapters have been displayed using 12 bit and 13 bit coding. Such coders can be expected to give SNR's of 63dB and 69dB, assuming linear encoders and speech-like signals. Companding encoders may be expected to give some 20dB advantage over these figures but have not yet been demonstrated. To measure a 65dB SQNR to within 0,1dB would require an 85dB test set back-to-back SNR; to measure within 1dB would require a test set SNR of about 71dB.

For telephonic use, it is reasonable to expect at best an eight bit companding system, even for tandem links with repeated encoding and decoding, (Ramsden). It is expected that such systems will proliferate as

the CEPT recommended European standard. An eight bit code, 13 segment A-law companding system gives a peak SQNR of over 38dB. Such a system has an SQNR of over 37dB over an input range of some -32dB to -3dB referred to peak input. It may be required to test different companding systems where this flat portion of the SQNR characteristic has been traded for a higher peak SQNR. Cattermole (1969, pp 135-138) has derived results for the variation of SQNR as a function of A, for A- law companding. An eight bit coding will give an SQNR of about 42,5dB for A equal to a value of ten. Greater deviations from the typical values of A around 100 are unlikely to be of general interest.

Hence the measurement requirement ranges from telephonic systems with SQNR below 43dB, to extreme fidelity systems with SQNR over 65dB. The measurement of SQNR's up to 43dB has been adopted as a minimum requirement for the present work. The aim, however, has been to seek to make the system proposed as useful as possible in the testing of higher fidelity systems and to determine the practical limits of this usefulness.

It is noted that the Marconi TF 2807 is specified to measure SQNR from 8dB to 42dB within an error of $\pm$ 0,5dB. There is some motivation for designing a more digitally oriented instrument even within these specifications. However, in the present work it has been sought to determine SQNR meaningful to at least $\pm$0,25dB over the same range. It is felt that a reading accuracy within 0,1dB is a worthwhile target, but the need for an accuracy significantly better than this has not yet been established.

On the basis of the specification established above, it is necessary to examine those aspects of a test system which influence the measured QD by even 0,1dB.

2.2 An Evaluation of Time Domain Subtraction

In the design of QD measuring apparatus, it is necessary to make an early decision as to whether a time domain or a frequency domain approach will be pursued. In the present work, it has been decided to pursue a frequency domain approach. This is a major decision in defining the system, and warrants a more than cursory examination of time domain systems.

Cattermole (1969, p 238) says of time domain systems that :

'It might, therefore, seem possible to measure it [SQNR] by direct comparison of input and output, supposing these to be available at one place. In general, this is not so. The constant delay and dispersion introduced by the baseband filters, makes direct comparison meaningless.'

Cattermole allows that it may be possible to conduct research studies by such means, but for a practical communications system concludes that :

'Thus we need a method of estimating the quantising noise without having the complete error waveform available.'

No analysis of time domain subtraction for QD measurement systems has been found in the literature. The analysis of the remainder of this section is presented because it guided the choice of a frequency domain approach in the present work. It has no pretensions to rigour.

A time domain subtraction scheme must present two signals to the subtraction device. One signal, of course, is provided by the PCM link under test. The satisfactory provision of a suitable reference version of the test signal, however, appears to be a difficult matter.

In order to analytically determine the general requirements for time domain subtraction systems, it is necessary first to develop a manageable but conceptually correct model of the test situation. Figure 5(a) represents a general PCM system as determined by its functional blocks. In this model, the $G_i(s)$ are transfer functions and s is the usual Laplacian operator.

$s = \sigma + j\omega$.

Immediately after the input to the PCM system there will, in general be an anti-aliasing filter. This is expected to be a fully analog stage in the sense of a linear time invariant continuous time and amplitude model. It may thus be correctly represented by some transfer function. The major part of the PCM system is, of course, a sampled data (discrete time) system. Nevertheless, by admitting exponential forms in the transfer function, it is possible to incorporate the sampled data nature in a transfer function model. This idea is detailed in Gupta (p 164 et seq). A zero-order sample and hold circuit can be represented as having a transfer function.

$$G_{oh}(s) = \frac{1 - e^{-Ts}}{s} \qquad (2.2\text{-}1)$$

where T is the sampling period.

In Figure 5(a), the anti-aliasing filter and input sampling circuit transfer functions have been lumped together and are denoted simply by $G_1(s)$. Similarly, any output hold circuitry and the output reconstruction filter have been lumped, and are denoted by $G_3(s)$. The output of the quantisation noise source is only very weakly correlated with the input signal.

There is also a frequency independent delay through the PCM link. This is not necessarily constrained to be an integral number of sampling periods, but is usually a constant for a particular link. This may be represented by a transfer function

$$G_2(s) = e^{-sT} \qquad (2.2\text{-}2)$$

for a delay of T seconds.

The gain compression and expansion non-linearities of the codec are not conveniently representable by transfer functions. They do, however, have time invariant amplitude transfer characteristics. In developing the models of Figures 5(b) and (c), it has been assumed that the compressor and expander have inverse characteristics, to a good approximation at least. This is consistent with the assumption that the QD is the major link impairment rather than total harmonic distortion (THD) due to mismatch between the compressing and expanding characteristics. Hence in developing Figure 5(c) from Figure 5(b), the compressor and expander together present a linear transfer characteristic to the input stimulus, and may be accounted for in $G_1(s)$.

The re-arrangement of Figure 5(a) to give Figure 5(b) has been done on the following basis. It is permissable to move the link delay block to after the expander because $G_2(s)$ is purely a delay and performs no signal amplitude modification. The delay may then be lumped with $G_3(s)$. The quantisation noise may be added after the expander, provided that the new model quantisation noise source Q_2 is defined so as to affect the system output signal in the same way as Q_1 does. As the expander is non-linear, Q_2 and Q_1 are dissimilar; the amplitude of the output of Q_2 is loosely dependent upon the amplitude of the compressed signal. The source Q_2 is the more directly useful source to measure for QD evaluation.

In simplifying the linear system of Figure 5(b) to give Figure 5(c), it is necessary only to define the noise source Q_3. This source is equivalent to the source Q_2 followed by the power spectrum shaping of $G_3(s)$.

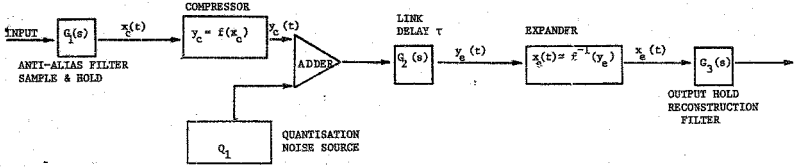


Figure 5(a) A Functional Representation of a general PCM System

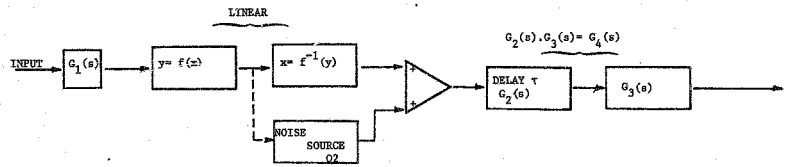


Figure 5(b) Development of a Simpler Model of a PCM System

Consider the model shown in Figure 5(c).

It is ideally required that the delay path be an exact replica of the PCM system under test, except that it should not include the additive quantisation noise source.

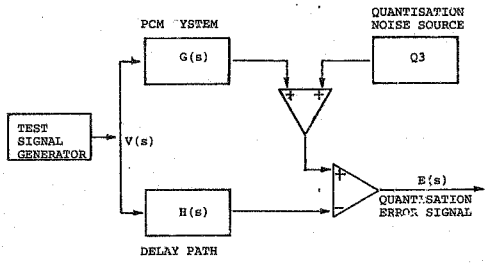


Figure 5(c) Analytic Representation of a Time Domain Subtraction System.

In order to derive the quantising error signal, it is conceptually required that $H(s) = G(s)$. A practical measurement will be made only to a limited precision. Hence we may consider that extent to which the above requirement may be relaxed.

The present work has considered only PCM systems intended for speech communication. A requirement has been the facility to test any such system. Hence $G(s)$ is restricted only in that it be the transfer function of a speech link. However, this does ensure that in practice there is one possible relaxation of the requirement that $H(s) = G(s)$.

Let $|G(s)| = A(s)$

and $\arg G(s) = P'(s)$ (2.2 - 3)

Then $P'(s) = P(s) + D(s)$ (2.2 - 4)

where D(s) is the linear phase component containing the pure delay in the PCM link.

For any speech link, the settling time of the impulse response will lie within a certain range. This range is limited because, for purposes of determining the test duration, D(s) may be excluded from the formation of the impulse response. Unless it is essential to save some tens of milliseconds in the test duration, it is reasonable to require only that $H(\omega) = G(\omega)$.

Let $|H(\omega)| = B(\omega)$

and $\arg H(\omega) = Q'(\omega)$ (2.2 - 5)

The signal subtraction requires that both

$B(\omega) = A(\omega)$ and $Q'(\omega) = P'(\omega)$ (2.2 - 6)

It might be sought to further relax these requirements by generating a test signal which is bandlimited to significantly less than 4 kHz (say). If this were done, the gain and phase matching would be required only over a range $2\pi f_L \leq \omega \leq 2\pi f_H$.

It would be acceptable in practice to relax the required equalities to approximate equalities. The requirement is the ability to assure the test instrument user that a measured signal to quantisation noise ratio is accurate to within a certain fixed uncertainty, say 0,1dB. Consider the general analytic problem of relating the error allowed in the gain and phase matches to the fixed overall uncertainty.

This analysis has to be in terms of fairly arbitrary transfer functions $G(\omega)$ and $H(\omega)$. It is necessary to define a quantitative performance parameter to indicate the degree of match achieved.

Let $V(\omega)$ be the frequency spectrum of the test signal. Let the error signal arising due to transfer function mismatch be $c(t)$, and the PCM sampling frequency be $f_s = \omega_s/2\pi$. $C(\omega)$ is the Fourier transform of $c(t)$.

Then

$C(\omega) = V(\omega)\,G(\omega) - V(\omega)\,H(\omega)$ (2.2 - 7)

Also, taking $|V(\omega)| = 0$ for $|\omega| < \omega_L$ or $|\omega| > \omega_H$,

gives the error power as

$$\int_{-\omega_o}^{+\omega_o} |C(\omega)|^2 d\omega = 2 \int_{\omega_L}^{\omega_H} |V(\omega)|^2 \cdot |G(\omega) - H(\omega)|^2 \, d\omega \qquad (2.2\text{-}8)$$

where $\omega_o = \omega_s/2$

Let the transform pair of the quantisation noise be $q(t) \longleftrightarrow Q(\omega)$.

The measured signal to quantising noise ratio would then be

$$SNR_m = \frac{\int_{\omega_L}^{\omega_H} |V(\omega)|^2 \, d\omega}{\int_{-\omega_o}^{+\omega_o} |Q(\omega) + C(\omega)|^2 \, d\omega + \omega_L} \qquad (2.2 - 9)$$

In order that the measurement lie within E dB (say), of the true value, it must be ensured at the very least that

$$10 \log_{10} \left\{ \frac{\int_{-\omega_o}^{+\omega_o} |Q(\omega) + C(\omega)|^2 \, d\omega}{\int_{-\omega_o}^{+\omega_o} |Q(\omega)|^2 \, d\omega} \right\} \leq E \qquad (2.2 - 10)$$

A major objection to time domain subtraction is that the above requirement is difficult to ensure. The quantisation noise power is, of course, not known. Furthermore, it is known to be a function of the test signal level, as is $C(\omega)$.

Let us consider also the task of implementing a time domain subtraction system. It would be necessary to provide the facilities for measuring $G(\omega)$ because $A(\omega)$ and $P'(\omega)$ are not known ab initio. This holds implications of complexity, cost, and long test duration. For a speech system there are reasonable limits to the minimum time in which a point on $G(\omega)$ can be evaluated as a steady state response. Additionally, it is neither desirable nor always possible to separate $G(\omega)$ from the associated quantisation noise. Then there is the non-trivial problem of measuring the phase characteristic of a system with geographically separated end points.

The measured values of $G(\omega)$ must then be used to program the synthesis of $B(\omega)$. This programming step could be a significant burden on the user, or involve additional circuitry/software. The facility for transfer function synthesis would have to be substantial one. Kikkert's system can at best synthesise a delay corresponding to a linear phase characteristic

$$Q'(\omega) = Q(\omega_1) + D(\omega) \quad , \text{ where } \quad \omega_L \leq \omega_1 \leq \omega_H$$

In practice it is possible to include a variable gain stage in the delay path. This may be set for each system tested to give minimum indicated quantisation error. This allows that $A(\omega)$ be known only to within a constant scaling factor, but again demands operator intervention or additional hardware. Kikkert's system cannot compensate for pessimism introduced by mismatch in $A(\omega)$ and $B(\omega)$.

The above analysis prompts a leaning towards narrowband test signals. Indeed, it is proposed that time domain subtraction schemes with narrowband test signals warrant more detailed investigation. Linear approximation of the amplitude and phase functions over a narrow band should be possible with sufficient accuracy. This would greatly reduce the problems of identifying and synthesising the transfer functions. However, the need to bandlimit the test signal even for time domain subtraction has not been recognised in the literature, and no detailed results are available for comparison with frequency domain subtraction techniques.

Even allowing for the bandlimiting stratagem, there are additional difficulties which must be considered. It must be anticipated that the subtraction device would need to be very precise and stable, and have good output slew-rate capability. This, however, is a mere technological consideration as opposed to a fundamental one.

A possible objection to time domain subtraction, however, is that two channels are required. This immediately doubles the effective test duration, ceteris paribus. Furthermore, it is apparently necessary that both channels be capable of transmitting in the same direction. This, in fairness, does not immediately exclude time domain subtraction from consideration where the requirement is the testing of a single bi-directional link with geographically separate end-points. It is as well to detail the considerations because such links are a common situation, and are implicit in the possible future proliferation of digital telephone subscriber loops.

Single bi-directional links may be tested by looping the link. This might be done by looping the digital portion of the link, and may be done at any convenient point. An alternative scheme may be considered for testing codecs in a telephone subscriber's instrument from the exchange: a command sequence may be sent from the exchange to set up in the subscriber's instrument a connection from the decoder output to the encoder input. Whichever of these two schemes is adopted, it is possible only to test the encoder-decoder pair at one end of the link. It is not possible to measure the performance of the encoder-decoder combination which exists in normal operation. However, the aim is either to develop a codec or to maintain it. In the former case, one seeks to measure the performance of a single unidirectional link, but the ends of the link are not widely separate. In the maintenance situation, one seeks merely to test a codec, and a loop test is probably sufficient.

A time domain system will be further complicated by the provision of a facility to test systems where the PCM encoded signal is stored for an

undetermined period of time. Such systems would, of course, include tape and disk recorders where playback is initiated manually. The requirement is that the test equipment incorporate the facility to synchronise the test signal and the decoder output signal. This is merely another factor to be weighed up in consideration of time domain as opposed to frequency domain systems, and again indicates a need for further detailed investigation of time domain systems.

In general terms, then, it seems that time domain systems involve complications of implementation. These complications stem from the requirement that the test signal be completely known at the receive end. Frequency domain systems, in contrast, require only that the amplitude spectrum band limits of the test signal be known at the receive end. Hence it was decided to pursue a frequency domain approach in the present work.

2.3 An Evaluation of Established Requirements for Frequency Domain Systems

The conventionally established requirements for frequency domain systems have been described in Section 1.4. It has been established in Section 2.1 that, for the accuracy sought in the present work, it is necessary to examine all the mechanisms which affect the SQNR measured by even 0,1dB or less. As was indicated in Section 1.4, the quantitative requirements of several of the test parameters have not been established precisely in the literature. The purpose of this section is to discuss further the test requirements, so providing a basis for the approach to precise parameter determination introduced in Section 2.6.

2.3.1 Frequency Spectral Requirements

The test stimulus is required to have certain attributes in terms of its frequency spectrum, and additionally to have certain statistical properties. Among the statistical properties, the amplitude probability density function (PDF) is of particular interest. The signal spectral requirements will be considered firstly; the statistical requirements are discussed in Section 2.3.2.

It is worth making a distinction between two groups of frequency spectral requirements imposed upon the stimulus. Certain requirements are consequences of the test method and the substantial correlation between the quantisation noise and the test stimulus. Other constraints are imposed on the test stimulus frequency spectrum in order that the required PDF may be achieved.

As noted in Section 1.4.2, there is no account in the available literature of a quantitative determination of SQNR as a function of centre frequency of the test stimulus. It is re-iterated that if this function fluctuates by 0,1dB, then it is necessary to make a reasoned selection of the stimulus centre frequency. This selection has been constrained by the test method used, in order to allow a significant noise measurement bandwidth. It is on this basis that Rolls and Webster have argued that the frequency should be 'in the lower end of the input frequency band'. In Section 2.4.2 a measurement technique is proposed which involves the injection of a signal near the upper end of the input frequency band. It is felt that a detailed evaluation of this test method is a necessary step in the design of a practical QD measurement set. It has been necessary throughout the present work to take full cognizance of this condition. While a facility for evaluating this test method has been designed into the hardware and software test systems built, no evaluation of the test method has yet been quantified. It is thus noteworthy that even if the constraint of the measurement method is removed, there is still cause for requiring that the centre frequency be in the lower end of the input band. The motivation is that more than half the long term mean power of speech is concentrated in the frequency band below 450 Hz. Richards (1973, pp 72-75) has given experimentally determined spectral data over the band 100 Hz to 8 kHz. He has noted that, for telephones, the full range is not transmitted and that, generally, 'all parts of the range do not have the same importance'. Accepting, then, a low end centre frequency, Cattermole (1969, p 237) has noted that noise power measurement may be hampered by slow beats when using a nominal 800 Hz tone and an 8 kHz PCM sampling frequency. It seems advisable to avoid a stimulus centre frequency which is a sub-harmonic of the sampling frequency, but in general the sampling frequency may be unknown initially.

There is a further factor which militates for the use of a low stimulus centre frequency. Cattermole (1969, particularly pp 175-182) has analysed quantisation noise effects due to small companding imperfections. He has noted that an arbitrary but significant distinction may be drawn between quantisation noise and harmonic distortion effects due to a displacement between compressor and expander (as shown in Fig. 6).

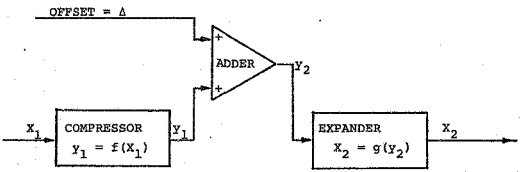


Figure 6 Offset between Compressor and Expander.

This distortion may exist even in the absence of quantising. Suffice it to say, however, that this offset is expected to be common in PCM systems; the offset is introduced where similar weighting mechanisms are used for both encoder and decoder. This results in a decoded level being at one extreme of the input quantum level, as opposed to the centre, but obviates complication of the decoder. Cattermole has derived an approximate expression for this distortion with sinusoidal excitation, and this allows adequately accurate computation for practical telephone codecs, both A-law and μ-law. The odd harmonic components are zero. The total distortion power is only 0,2dB above the power of the second harmonic alone. Cattermole gives a representative telephone example for a 7-bit codec with 25dB companding advantage: a displacement of $\Delta = -1/128$ gives a second harmonic 35dB down. The idea in the present work is that it is possible to exclude this second harmonic contribution from the measured SQNR, by

choosing a low test stimulus centre frequency. By excluding this contribution in one QD measurement on a system, and then re-measuring QD to include this component, it may be possible to identify the presence of an offset and to quantify its noise contribution. Consider, as a telephone example, a stimulus centre frequency of 420 Hz (say) as opposed to the usual 500 Hz. The conventional noise measuring filter with lower band-edge at 860 Hz would then suppress the second harmonic component. The possible motivation for excluding the second harmonic is that the harmonic distortion is distinguishable from quantisation distortion by its simpler correlation with the test stimulus; and the two effects are subjectively different. However, further investigation is needed to decide this question in terms of choice of stimulus centre frequency.

Referring to Figure 6 and the model of Figure 5(b), it is clear that non-linearities may arise due to mismatch of compressing and expanding characteristics,

i.e. when $g(Y_2) \neq f^{-1}(Y_2)$.

Cattermole (pp 183 - 192) has given some approximate analysis to derive asymptotic limits to the degradation of SNR due to this mismatch. He has pointed out that both systematic and random displacements of encoding and decoding levels must be considered, and this will incorporate the effects of circuit noise and imperfect encoder decision circuits.

Cattermole has defined conditions under which the various noise powers add independently, and holds that input noise and encoding imperfections may be 'indistinguishable' in practical measurements. The results are not detailed here because no way can be seen to relate this general analysis to the choice of test stimulus centre frequency in such a way as to enable the different impairment sources to be separated in measurement of QD.

The second frequency spectral requirement of the test stimulus is its bandwidth. The bandwidth must be small enough to be compatible with a low centre frequency and a reasonable noise measurement bandwidth. Yet the essential requirement is that the bandwidth and test duration be sufficient to allow the required amplitude PDF to be approached. The bandwidth-time law is not a restriction if a pseudo-random stimulus is used.

It is clear that there must exist an uncertainty principle which relates the minimum uncertainty in measured SQNR to, amongst other parameters, the measurement duration. No formal analysis of this has been encountered in the literature, and none has been felt necessary in the present work. The mechanism by which the test duration influences measured SQNR, for the test systems outlined, is again the limitation of achievable

PDF. It is necessary to ensure that the test duration exceeds the minimum corresponding to the 0,1dB SQNR uncertainty sought. This is covered in Section 2.6. With this minimum assurance, it is necessary to fix a maximum bound to which the test duration may acceptably be stretched to allow convenient engineering. Considering an automatic tester to do daily single level checks on 10 000 links, say, allows a test duration of over 8 seconds per link. Eight seconds has been chosen as the upper bound on test duration. This allows a full 30 point SQNR characteristic to be established in four minutes.

Reference has been made above to the test duration. The duration of signal and noise level measurements may nearly equal the test duration, because the settling time of a speech link is small. Once the measurement duration has been fixed, it is most convenient to make the test stimulus periodic with period equal to the measurement duration. Then no special synchronism is required between stimulus injection and receive end measurements.

A test stimulus of period T seconds will, of course, have a discrete frequency spectrum with amplitude spectral component spacing of $f = 1/T$ Hz. It is necessary to ensure that there is a sufficient number of spectral lines within the bandpass limits of the test stimulus, in order that the spectrum appears spread within these limits. A broader stimulus bandwidth together with more numerous spectral lines of significant amplitude will tend to give a flatter quantisation noise spectrum as required by the test method.

2.3.2 Statistical Properties Required

Together with the above frequency spectral requirements, it is necessary that the test stimulus has a certain amplitude PDF. The purpose of this section is to examine the question of what PDF the stimulus should be required to have. An implicit question here is that of the role of the stimulus PDF in determining the SQNR measured. The mechanism by which the PDF affects the SQNR has been satisfactorily determined, and Cattermole (p 126) has given a good brief analysis. The analysis is given immediately below as it is the starting point for the subsequent evaluation of what PDF the test stimulus should be required to exhibit.

Consider a quantiser with signal range divided into N intervals which are not necessarily equal. Let the i-th interval be of width q_i centred on amplitude x_i. Let signals in the range $x_i - q_i/2 \leq x < x_i + q_i/2$ be quantised to x_i. Let the amplitude probability density (PDF) of the signal be denoted p(x).

Then the squared error due to signals falling in the i-th interval may be seen in statistical average form

$$E_i^2 \approx \int_{x_i - q_i}^{x_i + q_i} (x - x_i)^2 \, p(x) \, dx \qquad (2.3 - 1)$$

In cases where the quanta q_i are small compared to the commonly occupied signal range and where the PDF $p(x)$ is a smooth function, it is usual to assume that $p(x)$ is approximately constant over a quantum interval and equal to $p(x_i)$. Noting that the squared error contribution is independent of the polarity of the error, the above approximation may be seen as approximating the $p(x)$ curve over each interval by a straight line segment. This approximation is not expected to be significant. Hence

$$E_i^2 \approx p(x_i) \int_{x_i - q_i}^{x_i + q_i} (x - x_i)^2 \, dx$$

$$\approx \frac{1}{12} \, p(x_i) \cdot q_i^3 \qquad (2.3 - 2)$$

The i-th interval will make the above contribution with probability

$$P_i = \int_{x_i - q_i/2}^{x_i + q_i/2} p(x) \, dx \qquad (2.3 - 3a)$$

Using the same approximation as before :

$$P_i \approx p(x_i) \cdot q_i \qquad (2.3 - 3b)$$

Hence summing the contributions of all intervals to the total squared error :

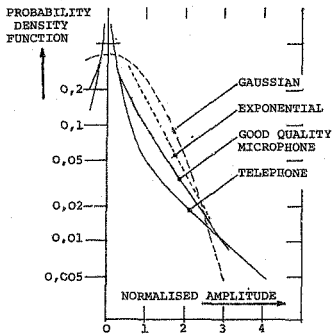
$$E^2 = \sum_{i=1}^{N} E_i^2$$

$$= \frac{1}{12} \sum_{i=1}^{N} q_i^2 \cdot P_i \qquad (2.3 - 4)$$

Thus the PDF weights the error power contribution of each quantum interval in forming the average error power. In view of this direct role played by the signal PDF in determining the SQNR, it is now necessary to examine the problem of specifying what the test stimulus PDF should be.

Waddington (1969) was constrained to generate a test stimulus by filtering a natural noise source which was both white and had Gaussian PDF. As an icipatad from linear filtering theory (Taub and Schilling, 1971, pp 240-242), the narrow band stimulus also had Gaussian PDF. Rolls and Webster have filtered noise from various linear feedback shift register sources.

These have given approximately Gaussian PDF's. They have formally introduced the requirement that the stimulus have Gaussian PDF, on the basis that the pseudo-random stimulus PDF should be as similar as possible to the PDF of Waddington's true noise stimulus. However, it must be emphasised that this is not a fundamental requirement. It is preferable to require that the stimulus PDF be similar to the PDF of 'average' speech. It has been convenient to use the experimental determinations of speech PDF given by Richards (1973, pp 63-69). These experimental speech PDF's are shown in Figure 7, together with the curves for elementary analytic models.



(a) Normalised amplitude = ratio *INSTANTANEOUS/RMS*
(b) Exponential distribution is a straight line here, because ordinate scale is logarithmic.

Figure 7 Long Term Average PDF's of Speech, and Analytic Distributions (after Richards, 1973, p 65).

From Fig. 7, it is noted that neither the Gaussian, nor the exponential distribution is a good fit to the experimental curves. The extent to which the apparent dissimilarity in these distributions is significant in the measured SQNR, is a central question in this discussion. This question is answered in Section 5.2. It is convenient to note now, however, that the gamma distribution suggested by Richards (1973, p 66) will closely fit the experimental curves for suitable values of the parameter h; the distribution may be written as :

$$p(x) = \frac{k}{2 \cdot \Gamma(h)} \cdot (kx)^{h-1} \cdot e^{-kx} \qquad (2.3 - 5)$$

where h is a parameter, and $k = \left\{h \cdot (h + 1)\right\}^{\frac{1}{2}}$, Γ is the gamma function, and x is equal to the instantaneous amplitude divided by the long term RMS amplitude of the speech signal.

p(x) is the probability density function.

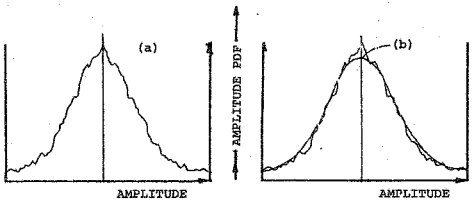
h = 0,5 gives a good fit to the curve for good fidelity speech, and h = 0,2 will model a carbon microphone signal.

A more approximate speech PDF model combines two equiprobable Gaussian distributions which are 23dB different in level. This is noteworthy particularly as the starting point of the test algorithm proposed in Section 5.2

All the speech PDF models considered in the present work have been assumed to be symmetrical for positive and negative amplitudes. In general, this assumption does not hold for any individual talker, but it is felt that departures from symmetry are not systematic and reflect detail unwarranted in the present work.

Accepting for the moment that the test stimulus should ideally have a PDF like the long term average power of speech, it is necessary to assess the acceptable limits to variations from this PDF. These limits cannot be precisely determined for a general purpose distortion tester, however, because the sensitivity of measured SQNR to input distribution depends on the particular coding and the companding law and its segmentation and imperfections. Cattermole has analysed SQNR characteristics with stimulus distribution as qualifying parameter. For unsegmented logarithmic A-law companding, his calculations show that the SQNR of a Gaussian signal may be up to 0,3dB better than that of an exponential signal. The calculations also show that for unsegmented μ-law companding, the exponential signal would have 0,3dB better SQNR than a Gaussian signal. Calculations using the gamma distribution have not been found in the literature, but the above

results serve to give some indication of SQNR sensitivity to signal distribution for common systems. The results of Rolls and Webster are similarly noteworthy. They have compared measured SQNR characteristics as obtained with a Gaussian true noise source, with those obtained with an approximately Gaussian pseudo-random source. In common PCM systems, the measurements differ in places by over 0,3dB. The extent of the approximation to the Gaussian PDF achieved by Rolls and Webster is shown in Fig. 8.



(a) PDF of filtered pseudo-random noise
(b) Sketch of true Gaussian PDF

Figure 8 Gaussian PDF Approximation Achieved by Rolls and Webster (After Rolls and Webster, 1974, p 94)

It is noted, too, that there is no consistent bias as to whether the true noise or the feedback shift register source gives the higher SQNR. Indeed, this is apparently PCM system dependent. Quantitative SQNR sensitivity to PDF shape is explored in principle in Section 5.2.

The above considerations have led to the following system requirements. The test stimulus PDF should be speech-like; in analytic terms it should approximate to the gamma distribution for parameter $h = 0,5$. The test signal may also be constrained in frequency spectrum as indicated in Section 2.3.1. Accordingly, no additional constraints can be imposed on the stimulus second order statistical properties.

A speech-like PDF has been proposed in preference to (say) a Gaussian PDF. Inherent in this preference is a desire to make a SQNR measurement which is more representative of the speech communication capability of the

link. It is important to note explicitly that the speech communication capability is only meaningful as a guide to the subjectively apparent overall quality of the path, and it is for this reason that one seeks to measure SQNR in PCM systems. Richards (1973, p 294) has suggested that for purely analog systems, at least, measurements of harmonic and intermodulation distortion 'do not, and are not required to, bear any direct fundamental relation to the subjective performance of telephone speech paths'. In the present work, consideration has been given to making the SQNR measurement *comparable to subjective quality assessments.* It is apparent that more careful and extensive analysis of the averaging processes is needed, so that good correlation can be assured between subjective and objective assessments (see Belcher, 1978).

The effect of a speech-like PDF, as opposed to a sinusoid derived PDF, is the smoothing of the SQNR versus stimulus level characteristic. The process is one of li[illegible] averaging, in that there is no level dependent weighting of noise or signal contributions to the averaging sum. This [illegible]y differ from a subjective assessment process. Waddington (1969) has noted that the minimum measured SQNR which is subjectively acceptable, is dependent on signal lev[illegible]n the range of -5dBm to -25dBm signals, a SQNR of 22dB is just acceptable. For a -42dBm signal, a SQNR of 16dB is easily acceptable because distortion products are below the threshold of audibility. The quantitative difference between measured and subjective assessments, arising because of this mechanism, has not been investigated. Similarly, no quantitative work has been done to justify the use of long term speech statistical averages. Richards (1973, p 442) has noted that 'speech cannot, for the present purposes, be treated as though it were a statistically stationary process'. This is an interesting fundamental *point, but daunting to investigate at this stage.*

Another subjective averaging process might correspond to that of psophometric weighting of the quantising noise in particular. This is a well established technique, (Richards, pp 248-254), and has not been explicitly cited in the following work. Nevertheless, it is a necessary feature of a QD tester, and has been a factor in motivating the system proposed in Section 2.5.

It may also be desirable to differentially weight quantisation noise as opposed to normal circuit noise. Richards has noted that, because of the correlation between stimulus and quantisation noise, the above noise types differ in subjective impairment of intelligibility.

The requirement of a non-Gaussian PDF makes formal analysis of the test system intractable, except possibly by the technique proposed in Section ...2. The difficulty is illustrated in Fig. 9.

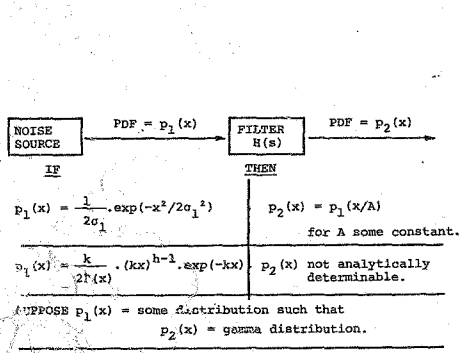


Figure 9 Possible PDF's in the Test Stimulus Generator

The noise source techniques described by Walker (1974, 1977) allow $p_1(x)$ to be set as desired. There are three broad approaches to generating sequences in which the second order statistics are controllable in addition to the PDF, including a technique recently described by Walker (1978). An early decision in the present work was to limit the investigation to Walker's earlier technique in which the second order statistical properties of the source are not controllable. The use of sources with prescribable correlation properties as well as prescribable PDF is discussed in Section 5.5. These sources do not offer immediate advantages. In considering Figure 9, it is thus necessary to consider that the second order statistical properties of the noise source are not controllable. In the system proposed in Section 2.5, transfer function $H(s)$ is not initially known. An easy approach to the solution of these problems is outlined in Section 7.6.

In the course of the present work, the existence of some $p_1(x)$ which would give

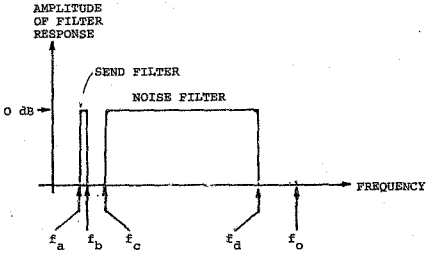
$$p_2(x) = \frac{k(kx)^{h-1}.e^{-kx}}{2.\Gamma(h)}$$ was assumed.

Pursuing the experimental approach of Section 2.6 has shown the assumption of existence of this $p_1(x)$ to be unwarranted for the noise source used. This point is discussed in Section 4.1 and in Section 5.1.

2.4 Measurement Techniques with Frequency Domain Subtraction

2.4.1 An Evaluation of the Established Measurement Technique

The conventional measurement technique has been outlined in Section 1.3.2. A convenient representation is shown in Figure 10, and this measurement technique will subsequently be referred to as single band injection.



f_s = PCM sampling frequency

$f_o = f_s/2$ = folding frequency

Guardband = $f_c - f_b$

True measurement bandwidth = $f_A - f_c$

Figure 10 The Single Band Injection Measurement Technique.

The selection of the parameters f_a to f_d in this technique has been discussed in Section 2.3.1. For reasonable parameter values, there are three disadvantages which are inherent in the single band injection technique. The technique will always give an optimistic measured SQNR, because the true measurement bandwidth is less than the folding frequency. This optimism may be reduced by a correction factor, giving

$$SQNR_{corrected} = SQNR_{measured} + 10 \log_{10}\left\{\frac{f_d - f_c}{f_o}\right\} \quad (2.4 - 1)$$

where the above SQNR's are expressed in dB.

Here, and in all subsequent work, the restriction of the channel below f_a and above zero frequency has been omitted for simplicity. This factor may easily be inserted as required for particular PCM systems.

The above correction factor is based on an assumption that the average quantisation noise power spectral density over the range zero frequency to f_c is the same as that over f_c to f_d. This is not true in general. Cattermole (1969, p 240) has measured a typical quantisation noise frequency spectrum using narrowband Gaussian noise centred on 500 Hz. Here the noise spectrum rolls off below 30 Hz at a rate of roughly 20dB/decade. It is generally expected that in the frequency band below the stimulus injection band, the quantisation noise average spectral density will be lower than that at (say) $3f_c/4$. Hence it is not attractive to try to ensure that this measurement bandwidth correction factor will be accurate to well within 0,1dB.

A second difficulty with the single band injection technique is that the injection bandwidth is severely constrained in practice. This may hamper the generation of a test stimulus with the required PDF. Certainly it is not possible to make the test stimulus resemble speech in the frequency spectrum.

The third disadvantage of single band injection is that the send and noise measurement filters must slope off very steeply so that leakage due to passband overlap is minimised. It is not possible to allow a large guardband between these two filters, because this would reduce the noise measurement bandwidth $f_d - f_c$.

2.4.2 A Proposed Measurement Technique

It is proposed that a more convenient measurement technique would be to inject first a low centre frequency stimulus, and then inject a high centre frequency stimulus. This technique will be referred to as dual band injection, and is illustrated in Figure 11.

It is felt that a quantitative evaluation of this technique is a necessary step in the design of a QD measurement set. It has been necessary throughout the present work to take full cognizance of this condition. While a facility for evaluating this test method has been designed into the hardware and software test systems constructed, no evaluation of the test method has yet been quantified.

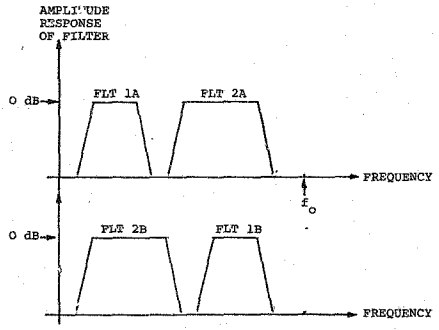


Figure 11 The Dual Band Injection Technique Proposed

The operation is as follows, assuming arbitrary notation. Filter FLT 1A injects a signal of level A dB for duration T_A, while part of the noise power is measured through filter FLT 2A. When this is complete, filter FLT 1B injects a signal of level B dB for duration T_B, while the noise power is measured through filter FLT 2B. The noise and signal energies may then be summed and an average SQNR calculated.

This technique allows many degrees of freedom in design. It permits of broader injection bandwidths, substantial guard bands and relaxed filter slope-off requirements. It will be seen that the use of two injection periods is a feature which fits in nicely with the testing algorithm proposed in Section 5.2. The passbands of the two send filters may be chosen so that the total test stimulus bears some loose resemblance to speech in the frequency domain also. The above factors are potentially

very substantial advantages of the dual band injection technique. Other characteristics of the dual band injection technique must be investigated in the light of the quantisation noise frequency spectrum.

The detailed quantisation noise spectrum in a particular case will be a function not only of the test system, but also of the PCM encoder. Hence a general purpose tester should anticipate, and be characterised for, various types of noise spectrum.

Bennett (1948) has given a theoretical analysis of the spectra of quantisation errors, but has detailed only encoders with uniform quantum size. This has shown that, for many quantisation intervals, the error spectrum is flat. Bennett has quoted experimental results which, while commonly exceeding the theoretically expected noise levels by 2dB, confirm that the error spectrum is flat. It is important to note that the analysis has been done for a Gaussian random noise source which has no energy outside of a band from zero frequency to frequency f_o. A significance of the normal noise assumption is that Bennett's analysis is possible only because the two-dimensional PDF of such a source can be determined (Rice, 1945).

The Gaussian random process is special in that all the statistical properties may be found from the autocorrelation function, and this is not true generally of random processes, (Harman, 1963, pp 81-83). Hence no attempt has been made here to extend Bennett's analysis for a source with the more realistic gamma PDF. A significance of Bennett's choice of stimulus band is that his experimental results demonstrate only that the quantisation noise spectrum is flat over frequencies above the stimulus frequency. The analysis, however, shows that there is virtually no correlation between the errors in successive samples except when there is complete correlation of successive stimulus values. This may be interpreted as suggesting that the error spectrum is essentially independent of stimulus band limit f_o.

Sripad and Snyder (1977) have analysed in detail the quantisation error statistical properties of linear encoders with Gaussian distributed inputs. They have derived a necessary and sufficient condition for quantisation errors to be uniformly distributed. They have proved that the noise will be white provided that the errors are uniformly distributed, this is a weaker requirement than that of no correlation between noise and input. Gaussian inputs do not satisfy the requirements for a uniform noise model. However, Sripad and Snyder have shown that the quantisation noise becomes less coloured as the ratio of the standard deviation of the stimulus to the quantum size increases.

On the above basis, it is expected that flat noise spectra may be encountered. It is noted that the analyses which lead to the anticipation of flat noise spectra, suggest that the noise spectrum is sensibly independent of the centre frequency of the stimulus band. This independence will be referred to as injection frequency insensitivity. In the case of a flat spectrum, injection frequency insensitive system, either the single or the dual band injection technique could meet the accuracy specification. However, the dual band injection technique would have the advantages listed previously, and these will be lumped together under the term 'implementation' advantages.

Cases of injection frequency independent systems with non-flat error spectra are philosophically less probable. If such cases did arise, the dual band injection technique would have both implementation and accuracy advantages. The need to guess the error spectrum does not arise with dual band injection into injection frequency independent systems. In such cases, the two basic injection cycles may be weighted equally to determine the average SQNR.

More probable spectra are those similar to the example measured in a companding codec by Cattermole (1969, p 240). These noise spectra are neither flat nor injection frequency independent. No general analysis has been found of the effect on the noise spectrum due to the codec expander characteristic, but it appears that noise is added more readily to the frequency band above the injection frequency than to the band below injection. The quantitative extent of this has not been determined. Generally, however, this introduces the requirement that the test stimulus frequency spectrum resemble the average speech spectrum. If this were a strict requirement, incidentally, the attractiveness of a time domain subtraction approach would be enhanced. Dual band injection is advantageous to the extent that speech high and low frequency bands are distinct in time. In order that dual band injection be utilised fully, it is necessary to determine the optimal relative weighting of the high and low frequency stimuli. This is part of the difficult problem of averaging parameters to give SQNR measurements equivalent to subjective assessments, as outlined in Section 2.3. A particular difficulty is that while there is fairly distinct time segmentation of speech, syllabic segments do not show the frequency spectral segmentation corresponding to the dual band injection technique. Even if shorter intervals are analysed, and Richards (1973, p 72) has suggested that 20-40 millisecond intervals give statistically stationary processes, it is expected that spectra will be (low-pass filtered) excitations with attendant diminishing higher harmonics.

Richards (1973, p 72) has given curves to illustrate that the long term average speech spectral shape is a function of talking loudly or quickly. The quantitative significance of this phenomenon as it affects the averaging of the low and high frequency stimuli in dual band injection is not known, but is expected to be second order (small). Nevertheless, subsequent test system proposals can cater for stimulus level dependent averaging, as this does not introduce significant overhead.

In view of the observation that more than half the power of speech lies in frequencies below 450 Hz, it is natural to wonder whether single band injection is not sufficiently representative of speech. However, it is desirable to produce a test system which is applicable to the testing of any type of voice transmission system regardless of encoding system. This is particularly so when the testing requirements of a general digital data switching exchange are considered. It is thus desirable to establish the requirement of eventual test system applicability to differential PCM (DPCM) systems, perhaps extending in the limit to delta modulation (DM) systems. Mc Donald (1966) inter alia, has investigated the signal to noise ratio performance of DPCM systems with linear feedback networks. He has derived, by computer simulation, frequency spectra of the quantisation errors. The shapes of these are very strong functions of the quantiser loading (see Fig. is clear that the flat spectral assumption of the single ban on technique would lead to very substantial measurement errors. Furthermore, suffice it to note that slope overload effects may cause substantial encoding errors, so that a representative high frequency stimulus component is needed to establish these.

The error spectra given by Mc Donald (1966, p 1138) lend weight to the argument that it is not possible to measure SQNR to within even 1dB if gross assumptions are made about the error spectrum. It is unlikely that error spectra of PCM systems are substantially flatter than those of the DPCM systems studied by Mc Donald, because the DPCM error spectrum is not flat even with light loading. The spectra are reproduced from Mc Donald's paper in Fig. 12.

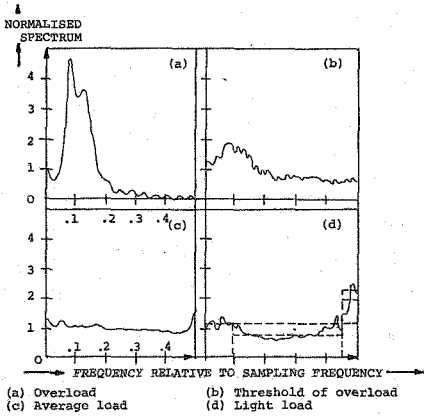


Figure 12 Error Spectra for a Simulated DPCM System (After Mc Donald, 1966, p 1138)

Let us consider the application of a single band injection technique to the measurement of the noise spectrum shown in Fig. 12(d). Assume that the noise power is measured through an ideal rectangular filter with normalised passband 0,1 to 0,45 relative to the sampling frequency. The estimated average power densities over the various intervals have been drawn in the figure as horizontal dotted lines. Using the graphically estimated data, it is clear that correcting the measurement bandwidth on the basis of a flat error spectrum will give a SQNR which is some 0,8 dB optimistic.

Mc Donald has used 5 seconds of tape recorded speech as the system stimulus. This signal has a frequency spectrum similar to average telephone speech and a PDF which is close to the gamma distribution. Mc Donald has motivated the use of this stimulus on the basis that it is likely to give representative second-order statistical properties. This type of stimulus cannot be easily used in a frequency domain subtraction scheme, but dual band injection may slightly alleviate this shortcoming.

Hence it has been felt that a fair evaluation of possible frequency domain subtraction systems must include an evaluation of the dual band injection technique. The system proposed in the next section meets the requirement that it be suitable for the implementation of a dual band injection technique if required. The design procedures outlined in Section 2.6 have taken cognizance of the need to evaluate the dual band technique, as have the simulators described in Chapter 3.

2.5 Outline of the Proposed Test System

The test system which has served as the working proposal in the present investigation is outlined in Figure 13.

The above system may be used as a basis for implementing any of the frequency domain test schemes described in previous sections. Many of the system functions in Fig. 13 are performed digitally. It may be assumed that the various test schemes are distinguished in implementation only insofar as these digital sections are different. The various schemes may differ only in software.

As has been suggested in previous sections, there are presently insufficient quantitative grounds to allow reasoned selection of a particular test scheme. The purpose of the system proposed in Fig. 13 is to help in the selection of a test scheme and in determination of design parameters in that scheme. The system is thus an experimental one.

The remainder of this section is intended to establish the functions and requirements of each part of the experimental test system.

2.5.1 The Pseudo-Random Noise Source

A (pseudo-random) noise source functional block has been shown in Fig. 13. When this label is used subsequently, it will refer to that functional block only, and not to the whole pseudo-random analog signal generator.

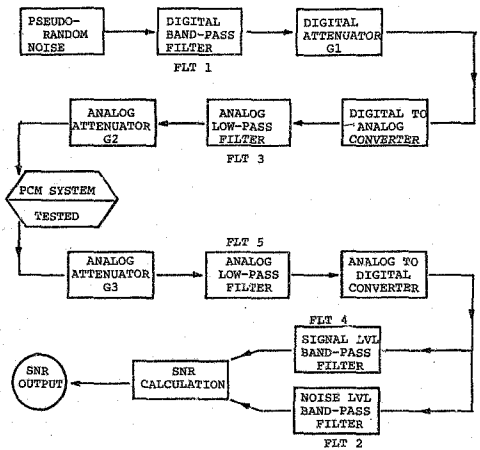


Figure 13 Functional Blocks Comprising an Experimental Test System

The (pseudo-random) noise source is required here to allow prescription of the noise PDF. No severe limitations as to the range of prescribable distributions were acceptable initially, and it is necessary that the noise conform accurately to the prescribed PDF. The noise source used in the present work has been one based on the techniques described by Walker (1974, 1977). These techniques permit of a fast and accurate source, and are not unduly constrained in the range of PDF prescribable. An aim of the present work has been to investigate the usefulness of Walker's techniques in QD measurement, so the selection of a noise source technique has been qualified by this constraint. This choice is not a critical one, as will be seen in Section 4.1.

The prescribed distribution will be referred to as the required PDF, and the actual noise source distribution as the achieved PDF. It will often be convenient to loosely refer to discrete probability distributions as PDF's.

The need to arbitrarily prescribe the required PDF was seen to arise in the following way. It is desired that the noise source achieved PDF be such that the filtered noise stimulus has prescribed speech-like PDF which is non-Gaussian. It is known that a Gaussian noise signal applied to a linear filter will produce an output which is also Gaussian, (Taub and Schilling, 1971, pp 240-242). No general theoretical results were found to relate a non-Gaussian input PDF $p_1(x)$ to an output PDF $p_2(x)$ for a transfer function H(s). However it was assumed that, for a particular H(s), there would exist some $p_1(x)$ such that $p_2(x)$ would be the desired speech-like distribution. This was illustrated in Fig. 9. Hence it was intended to search for this $p_1(x)$ by prescribing a variety of required PDF's, examining the filtered PDF of each, and so on. This approach has been explored and the findings are given in Section 4.1. The existence of suitable $p_1(x)$ has been shown to be an unwarranted assumption for the noise source used.

2.5.2 Filters

The critical filtering is performed by the send and the noise measuring filters, FLT 1 and FLT 2 in Fig. 13. It has been convenient to make the signal level filter FLT 4 similar to FLT 1. These three filters are all bandpass types. They differ in band edge frequencies and these frequencies were not initially determined. Also, the passband of each filter function needs to be shifted for different measurement cycles when dual band injection is used. Hence it has been most convenient to implement these filters digitally in the experimental system. It is envisaged that in eventual fully developed testers also, these filtering functions would be done digitally for reasons of high performance, reproducibility and stable characteristics. A further compelling reason for implementing FLT 2 and FLT 4 digitally is the convenient follow-on to digital true RMS power measurements, SNR calculation and display or output.

The stimulus generator up to the digital to analog stage is, of course, a sampled data system. Filter FLT 3 is the reconstruction filter and it has been felt wise to limit its function to that only. The PCM system under test may contain an anti-aliasing filter, but this cannot be relied upon in general to perform the function of FLT 3. There are some reasons to include FLT 3 in the test set rather than relying upon anti-aliasing filters in the system to be tested. Firstly, PCM systems intended

for signals which are known to be bandlimited by other considerations, may not incorporate anti-aliasing filters. The requirements for FLT 3 will be seen to be modest (Section 4.6), and restricted test set applicability would not be cost justified by the exclusion of FLT 3. Also, it may be necessary to test systems like PCM disk recorders and other good fidelity systems. These systems may incorporate anti-aliasing filters, but have sampling rates well above those acceptable for the test set. Test set sampling frequency requirements are discussed in Section 5.6.

An important experimental simplification rests upon the observation that FLT 3 is unnecessary if the test system sampling frequency is locked to that of the PCM system, and FLT 3 is assumed to be an ideal rectangular lowpass filter with cut-off frequency (say) at half the sampling frequency. However, as tester development proceeds, it is necessary to incorporate the envisaged FLT 3, in order to assess its harmonic distortion, noise, and modification of the test stimulus PDF.

2.5.3 Attenuators and Data Converters

It is necessary to be able to set the stimulus signal level accurately. The performance specifications of the Marconi TF 2807 in this regard have been adopted in the present work, and these data are given below to indicate the severity of the requirements.

0 dBmO reference level : -16dBm to +2dBm in 0,2dB steps

Send level : -60dBmO to +10dBmO in 1dB steps

The send level is set in the proposed system by digital attenuator G1 together with analog attenuator G2, as in Fig. 13. The aim has been to maximise the analog attenuator step size. However, in order to maintain the test set back-to-back SNR, there are trade-offs to be made between the relative amounts of digital and analog attenuation, and the required digital-to-analog (D/A) converter resolution.

A variable gain stage, say attenuator G3, is required to adjust the RMS value of the PCM system output before it is applied to the analog-to-digital (A/D) converter. It may be anticipated that the A/D converter would be one with constant quantum size, because it is required to be both fast and accurate. Also, the use of a linear encoding characteristic simplifies the subsequent filtering operations. It is desirable that the RMS input to such a linear A/D converter be 12dB or more down on the peak possible input (Bennett, 1948, p 450). Cattermole (1969, p 128) has suggested that peak clipping is negligible for a speech-like signal 20dB down on the overload limit. Hence it is not desirable to reduce the A/D converter RMS input below this. Similar restrictions apply

to the D/A converter. This is because the data converter resolutions will be seen later to be in the 12-bit or better class, so that an extra step of analog attenuation may become cheaper than an extra bit of resolution.

It is clear that the D/A converter will add its own broadband quantisation noise to the test stimulus, as well as signal components due to its imperfect linearity (relative accuracy). Similar noise components will be injected by the A/D converter. An approach to quantitatively evaluating these effects for design purposes will be outlined in the next section.

2.6 A Test System Simulator

2.6.1 The Purpose of Simulation

It has been suggested in the preceding sections of this chapter that there is a need for more quantitative descriptions of many fundamental aspects of quantisation distortion measurement. These aspects include, at the system level, the test technique and many parameters of the test stimulus. At a level closer to the design of a useful tester, there are similarly many parameters to be fixed. In the system proposed in Section 2.5, for example, it is necessary to fix the required filter orders, D/A and A/D converter precisions, digital and analog attenuator trade-off, and noise source parameters. Formal analysis of these systems is intractable, particularly for non-Gaussian signals. The analytic results which have been quoted from the literature are all too approximate.

Consequently, the present work has been concerned with the construction of a computer model from which representative quantitative results could be obtained. As noted, the system proposed in Section 2.5 provides a suitable basis for this model. This general purpose test system simulator has proved to be a very powerful tool. It is described in Chapter 3, and some results are given in Chapter 4.

The use of a computer simulation as the design tool here is most fortuitous in that a large portion of the proposed test system is digital in nature. Hence the realisation of a developed design from a simulation is a direct process. Similarly, it has been possible, by slight extension of computer hardware, to use the simulator to provide a laboratory QD test facility with general applicability.

In simulation, the approach has been to determine several parameters by observing the convergence of measured SQNR values as a function of those parameters. Clearly this process is PCM system dependent. Hence a PCM system was selected to serve as a typical example. The system chosen has

been built in hardware and modelled in software.

The hardware and software approaches are seen as complementary. The software simulations are more useful for initial investigations of a wide range of systems. However, it is difficult to model actual PCM encoders accurately, primarily because of the anti-aliasing filters. When particular versions of the proposed test system have been developed, it is desirable to do comparative tests with (say) the Marconi TF 2807 and PCM codecs in service. Also, an evaluation of a developed version of the proposed system must include the effects of imperfections in the analog lowpass-filter (FLT 3), and the D/A and A/D converters. Again this is more easily done by hardware measurements. The modification of signal PDF by analog filter FLT 3 must also be examined, and this can be similarly facilitated by hardware.

2.6.2 Hardware Considerations

It has been suggested, in Section 2.6.1, that there will arise in the course of development of the system proposed in Section 2.5, a need for a simulation of that system, and also a simulation incorporating in hardware the analog elements of that system. The purpose of this section is to outline the ideas which defined the particular hardware system design. This design has been implemented, and details are appended for reference (Appendices A,B,C). However, it is necessary here only to establish the operating environment for the software to be described in the next chapter.

It is convenient to have the facility to run on the same computer either the fully software simulation or the simulation incorporating analog hardware. The general purpose mini-computer used in the present work is a Nova 1200 manufactured by Data General. This machine was 'chosen' because of availability, but is quite suitable in that it is readily interfaced to other hardware. The Nova has 16 kilowords of core store, and this has been a major constraint in writing the simulation package. Peripherals include a dual floppy diskette drive and these are driven through the Decision Disk Operating System (DDOS). It will be seen in Section 3.2.2 that various DDOS subroutines have been linked into the assembly language interface between the simulation program and the special purpose tester hardware.

An essential item of the hardware test scheme is the digital to analog converter. The specification of this functional block was approached in the following manner. The D/A converter requirements are defined by the amount of quantisation noise that can be accepted from the D/A. In order

to calculate the D/A converter SQNR, it is necessary to determine the signal PDF. The D/A signal is, of course, the signal input to the analog filter (FLT 3 in Figure 13). The PDF of this signal is determinable only insofar as it corresponds to a particular analog filter output PDF. The filter output PDF's were envisaged to include speech-like distributions other than the Gaussian. For the Gaussian output PDF case, the D/A signal has a Gaussian PDF. For non-Gaussian outputs of the analog filter, the D/A converter signal PDF cannot be determined in general. The lack of general analytic techniques to relate filter input and output PDF's has been demonstrated by Davenport and Root (1959, p 190). Indeed, it is this established lack of analytic technique that has most strongly motivated the simulation approach in the present work. All that can be done to specify the PDF of the D/A converter signal is to consider only the Gaussian case. D/A converter specification on the basis of adequate performance for Gaussian signals has been found to be acceptable in the light of the results of Section 4.1. It is hence assumed that the D/A converter signal is Gaussian. Consequently, for any fixed D/A resolution, it is preferable that the D/A be a logarithmically expanding type with non-uniform quantum size. Available converters of this type, however, have SQNR figures comparable to those of the PCM systems to be tested. The requirement stands, however, and future test set designs may consider expanding D/A converters.

The presently available technology has limited the test set D/A and A/D converters to types with uniform quantum size. The simulator described in Chapter 3 has been used to verify that 12-bit data converters are just adequate, as independently anticipated by lengthy approximate calculations. It may be anticipated that the data converters for a realistic implementation of the system shown in Fig. 13 would be in the 14-bit class. Because limitations of present component technology constrain the data converters to types with uniform quantum size, this limitation has been built into the simulation programs. A 12-bit hardware D/A converter resolution was thus considered a specification consistent with the needs at the present stage of tester development. Such a converter was affordable, and has been interfaced to the Nova as detailed in Appendix B. A 12-bit A/D converter facility was already available on the Nova.

A hardware PCM system has been constructed. As mentioned previously, this PCM system has the purpose of allowing comparative tests to be done on a typical PCM system by developed and extant testers. In designing the

PCM system, however, the aim has been to make the system as flexible as possible, so that it might be of more general interest as a laboratory instrument.

It is a noteworthy simplification in simulation to note that the SQNR of the PCM system used is a fairly slowly changing function of the speech-like stimulus level. This allows the test system data converter quantisation noise sources to be lumped, and only an equivalent D/A resolution need be specified in the simulations. This idea has been carried over to the hardware system. Here the PCM encoder output can provide the digital data read into the Nova for the receive side digital filtering. The 12-bit A/D converter is then not initially necessary for studying the performance of developments of the proposed system incorporating the analog filter FLT 3. The PCM encoder may also be used to build up a totally satisfactory picture of the stimulus PDF after the analog filter FLT 3. An advantage of this approach is that the behaviour of the quantisation noise is not masked by decoder noise and imperfections, and it is useful in studying quantisation noise phenomena as opposed to test system performance. The PCM system is detailed in Appendix A.

It has been necessary to examine signal PDF's at various points in simulations, and to compare the PDF's achieved with those required. It was felt preferable to plot curves of these PDF's for visual comparison, rather than using (say) a numerical least squares error technique for curve matching. As many such curves were anticipated, the plotting procedure has been mechanised by interfacing a Hewlett-Packard 7005 B analog X-Y plotter to the Nova. The dual D/A converters and interfacing circuitry are detailed in Appendix C.

2.7 A Summary of the Ideas Defining the Simulator

This chapter has established the current performance requirements of a QD measurement set. The task of defining a test system to meet these requirements has been approached by firstly evaluating both the possible approaches to QD measurement - time domain subtraction and frequency domain subtraction. These detailed and explicit evaluations have not been reported in the present literature, and they establish a formal philosophical basis for QD measurement. On this basis, the decision to adopt the frequency domain approach has been motivated. The measurement technique currently used in a frequency domain approach has been evaluated, and an alternative measurement technique has been proposed. It has been suggested

that an evaluation of this dual band injection technique is a necessary step in the design of a QD measurement system. A proposed test system has been outlined in terms of circuit function. The proposed system is largely digital in nature and is inherently flexible enough to incorporate dual band injection if required. It has been shown that analytic evaluation of QD measurement is not readily feasible. The use of a test set/PCM system simulator as a design tool and development aid has been proposed. The hardware requirements to enable the simulator to be used for laboratory QD measurements on hardware PCM systems have been outlined.

3 TEST SYSTEM SIMULATION

3.1 Software Description

This section outlines the software package written to allow simulation of the various quantisation distortion test systems and to perform experimental SQNR measurements on hardware PCM systems. The aim of this section is to establish the purpose of each major program section, and so to define the program design requirements. Hence it will be sought to identify and motivate the more important program design decisions relating to structure; but it has also been felt necessary occasionally to comment on parameter values where decisions have rested on approximate numerical results. Detailed descriptions of software have not been given because many of the programs are written in a high-level language, and commented versions of these are given in Appendices D and E.

3.1.1 The Two Simulation Packages

An early, and important, software design decision was the choice of programming language. The Nova operating system (DDOS) supports two high level languages, Fortran and Basic. Fortran has the advantage of execution speed, but has hitherto been regarded as a poor development language for this type of experimental system because of the need for recompilation upon any program modification. Fortran has the further disadvantage that it has hitherto been a relatively complex matter to pass many variable values to the callable assembly language driver routines which interface to the peripheral hardware like the data converters. However, DDOS supports a version of Basic modified to greatly ease development and use of callable drivers, and this version is called CSBASIC. Nevertheless, it was decided to write the body of the simulation package in Fortran. The most compelling motivation for this choice is program execution speed, but Fortran also permits of large variable arrays so that representations of sequences of signal samples need not be artificially segmented as they would be in

C5BASIC. Another compelling factor is that Basic does not readily allow multiple precision variable data (Extended Basic User's Manual, p2-1) and it will be seen particularly in Section 3.1.5 that some 64 bit data representations are an inescapable requirement. Also, substantial economies of storage are possible by use of integer type data in Fortran, and the necessity for these economies will be seen in subsequent sections. Suffice it to say in retrospect that the present simulation package would probably not have been possible if written in C5BASIC for the Nova.

It has thus been necessary to alleviate the apparent difficulties of using Fortran for this application, i.e. frequent recompilations and complex callable driver interface. A further significant factor is the limit of 16 kilo-words of core storage on the Nova available for the present work. An unorthodox scheme has been devised to effect the Fortran to peripheral hardware interface. This scheme is outlined in Section 3.2.1, but it is emphasised that the scheme is conceptually and operationally simple. It is also reliable and allows more variable values to be passed than the conventional procedure, and so may be of more general interest than the simulation package. The difficulties of recompilation and severely limited core have been handled by devising a suitable structure for the program package.

The simulation package structuring concept is shown in Fig. 14. The simulation package is segmented into program sections representing the functional blocks of the test system. It has sometimes been convenient for reasons of size to segment program sections within a functional block, or to introduce segments as parts of the simulation procedure rather than of the test system. All such segments, however, have the structure shown in Fig. 14; they operate as distinct programs and will be called simply 'programs'. A program operates on one or more disk files (as input) and produces (as output) one or more disk files. The simulation package consists of a network of such programs, connected by disk file 'nodes'.

The execution path through the network is controlled by the simulator (human) operator. When a program has completed execution, the operator can initiate the execution of any program 'link' via DOS.

This package structure has several clear merits. The simulator may be made extremely flexible in operation. Yet there is no need for tedious interactive question and answer programs and sessions in order to elicit the operator's requirements in respect of the many options.

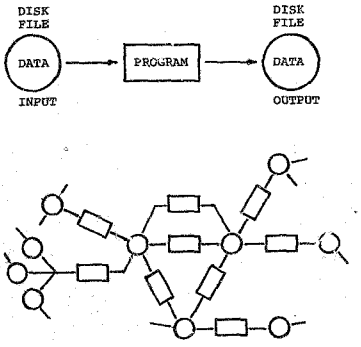


Figure 14 (a) The Basic Element of Simulation Package Structure on the Nova Computer.

(b) Disk File Nodes and Program Link Structure.

The structure also ensures very distinct and logically defined program modularity. Its merits in this regard must be compared with those of a structure having an executive program calling various subroutines. Fortran subroutines under DEOS are separately compiled and assembled to the relocatable binary stage. The main program and all subroutines must, however, be linked together. Hence such a scheme would necessitate the relinking of the whole package every time a programming modification were made -- and this can be a time-consuming operation. The structure proposed in Fig. 14, however, requires only the relinking of that program module in which the change was made. Hence the tedium of re-compiling, assembling and linking a large Fortran program under development has been substantially alleviated.

A further advantage of the proposed structure is that it builds a virtual memory scheme into the programming. This allows the large system to run in 16 kilo-words of core storage. The simulation package involves

both extensive code as well as large variable arrays. The proposed structure ensures that core is occupied at any one time only by those code segments which need to execute together. Similarly, variable arrays have been implemented which are larger than the core size. Small groups of variable data are read from the input disk file, processed, and written to the output disk file.

Implicit in the above discussion is an assumption as to the requirements of program execution speed. For the fully software simulations, clearly no real-time requirement exists. For simulations involving measurements on hardware, however, there is one real-time program link. This is illustrated in Fig. 15.

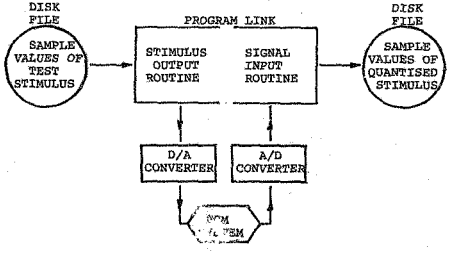


Figure 15 The Real-Time Program Link for Hardware Measurements

The nature of this link is detailed in Section 3.2. Other than this link however, there are no apparent real-time requirements, ab initio. Nevertheless, it has been found necessary to regard program execution time as important. This importance arises because it has been found that a single execution pass through a typical simulation may take up to some 15 hours execution time on the Nova. Failure to appreciate the magnitude of computational effort involved in test system simulation would severely restrict the usefulness of the simulator.

It should be emphasised, however, that simulation execution times of the order of hours do not necessarily imply that such measurement times would be fundamental to a test system developed using the simulator. The speed of a completely developed tester is determined by its hardware and firmware structures. These could be dedicated structures. In order to assess the minimum measurement processing time, it is necessary first to speculate as to these structures, and this is done in Section 5.4.

It is necessary to examine more precisely the role of the program package structuring in determining probable execution times. Firstly, it might be expected that long execution times would arise with the proposed structure due to the overhead of reading and writing many and large disk files. However, it has been found that disk overhead is an acceptably small portion of the execution time. The bulk of the execution time is, in fact, absorbed by the three digital filter programs. With just adequate filter order, each of these programs may run for over two hours.

A conclusion is that the proposed structure is much preferable here to an executive program and subroutines. This is because it is inherently easy to re-use previously computed partial results on successive simulations. For example, once a sampled test sequence has been stored in a disk file, no recomputation is necessary for subsequent simulations to examine various receive filters.

Another conclusion is that, for the initial and purely software simulations, it is certainly worthwhile to be able to run these simulations on a more powerful computer. It has been anticipated that the fundamental investigations, and even the development of a tester system would entail rather more simulations involving software only than those involving hardware also. Hence, in order to extend the usefulness of the simulation package previously outlined for the Nova computer, the purely software simulation facilities have been written into a package designed to run on an IBM 370/158 computer.

The simulations will be referred to as the Nova and IBM packages. A representative simulation might run for over 15 hours on the Nova. The equivalent IBM simulation would run in just under 6 minutes processor time. The IBM package also gives slightly more precise numerical results.

In order to realise this speed benefit, and to take advantage of the greater core storage available on the IBM 370, the IBM package structure differs from that of the Nova package. The disk files have been eliminated, and the structure shown in Fig. 16 has been used.

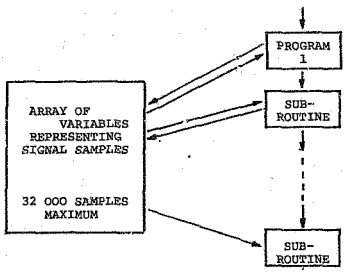


Figure 16 Elements of Simulation Package Structure on the IBM 370

The idea has been to have a single large array of signal samples in core, and the entire array notionally remains in core for the entire execution time. Each of the various program sections then operates on the array in an 'in-place' fashion, leaving a processed array for subsequent processing.

Because compilation and linking are no burden on the IBM 370, many of the program sections have been included as subroutines. An effort has been made to keep a direct correspondence between Nova program 'links', and IBM subroutines. The correspondence is sufficient, so that the intention is to outline the purpose and design of only the Nova programs. Listings of both packages are given in Appendices D and E.

The IBM package must run under the OS batch job facility, because the WITS system cannot provide the processing power. Consequently, the IBM simulator operator cannot conveniently execute the equivalent of a Nova program link, then think, and select the next link to be executed. This, together with the more extensive use of subroutines, has led to the IBM package being closer to the structure shown in Fig. 17.

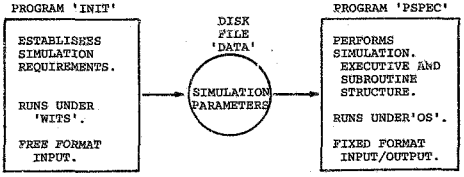


Figure 17 Operation of the Simulation Package on the IBM 370

The software simulation results quoted in Chapter 4 were obtained from the IBM package, (except where specifically noted). The IBM package is relatively straight-forward, and its operation will be apparent from the subsequent description of the Nova package.

3.1.2 Software Modules in the Nova Package

Figure 18 shows an outline of the Nova software. This serves to define program and disk file names and links, and will be referenced frequently in the remainder of this chapter where the purpose and design of each will be established.

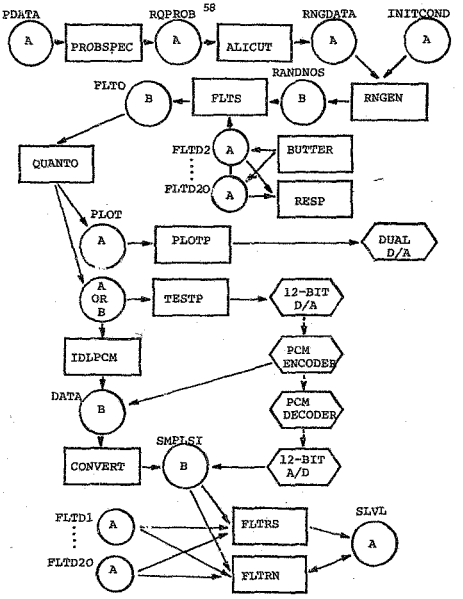


Figure 18 Nova Software Skeleton

The type of data representation in each disk file is indicated in Fig. 18. The policy has been to store data in ASCII form wherever it is possible that the simulator operator may wish to examine or edit the file. Type ASCII files may be examined via the DDOS COPY command. Similarly, minor modifications may be conveniently made via the DDOS edit facilities. Sequences of signal samples are unlikely to be meaningful to the operator, and as these are extensive, they have been stored on disk in binary form.

The storage saving is substantial; a single precision real variable requires four (8-bit) bytes in binary form and (say) 16 bytes in ASCII form.

Even with the use of binary input/output, all the software outlined in Fig. 18 cannot be accommodated on one user disk. The structure allows that segmentation of the basic software skeleton onto several user disks may be done at any convenient nodes. The (DDOS) systems disk is nearly filled by DDOS files, but may be used to connect two user disks. The disk file node common to the two user disks is created on the systems disk. In order not to unduly constrain the number and extent of the common user files on the systems disk, a few unrequired DDOS files were removed from the systems disk.

3.1.2 1 A Flexible Program to Aid Specification of Arbitrary PDF's: PROBSPEC

The pseudo-random noise source has been incorporated in the simulation in such a way as to allow operation with a large number of random variates (up to 1024 initially). The necessity for such apparently precise noise is examined in Section 3.1.2.3. In setting up the noise source, it is necessary to specify the relative frequency with which each variate is required to occur. This is the target or required probability distribution. It is desirable to obviate the tedium of specifying each point in the target distribution via keyboard input/output (I/O), but it is undesirable for the specification mechanism to constrain the range of target distributions.

A program called PROBSPEC has been written to meet these requirements. The operator may specify the number of points, up to a value of 101, that he wishes to explicitly specify on the target distribution. He then enters the co-ordinates of these explicit points, and the number of random variates required. PROBSPEC forms an approximate picture of the target distribution by constructing straight line segments to connect the specified points. The target probabilities of the random variates are computed, and written to the disk file RQPROB.

The initial data for target distributions come from published curves of speech PDF, and these are continuous. These curves may thus be sampled graphically at convenient, and not necessarily equidistant, values of abscissa; and the data entered to PROSPEC. Any measurement units are suitable as only the shape of the distribution is to be specified.

The sum of the computed target probabilities is expected to be unity. An important feature throughout the simulation is the use of normalised integer representations of fractional probabilities. Integer magnitudes

are represented on the Nova by 15 bits, and integers may thus have values up to 3 . The fractional probabilities are mapped into the representable intege ugh multiplication by 32768. Hence the sum of normalised probabi s in the computed target distribution should be 32768, and this is necessary for the proper operation of the subsequent program ALICUT.

Small numerical errors arise in the computation and normalisation of the target distribution. The value of this error is indicated to the operator, and he may elect to allow an algorithm to restore the target normalised distribution sum to the required value. This algorithm adjusts the target probability of each random variate by an integer value of three, starting with low absolute valued variates and continuing until the residual error is less than three. The residual error is brought to zero by an appropriate final adjustment, and the target distribution written to RQPROB. Alternatively, the operator may elect to correct the error by editing the disk file RQPROB under DDOS.

The co-ordinates of the points on the target distribution which were explicitly operator specified are stored in disk file PDATA. Local adjustments to the target distribution may be made easily by editing PDATA, and this file may be used to specify the target distribution when PROBSPEC is run subsequently.

There are two obvious methods by which the speech-like distributions may be generated using Walker's techniques. These are outlined in Fig. 19.



(a) Asymmetry possible, offset subtracted from generated sequence.

(b) Symmetry assumed, sign bit generated independently.

Figure 19 Approaches to Distribution Generation

It is no significant limitation to assume the target distribution to be symmetrical for positive and negative amplitudes. It has been found advantageous in the simulator to use the method of independent equiprobable sign bit generation, as is possible because the Markov source is of order zero. The advantages are that it is necessary to compute and store only half the number of target probabilities and alias and cut-off values for the pseudo-random source. Only the positive half of the distribution need be indicated to PROBSPEC.

It has been found most convenient in programming to constrain the total number of random variates required (N) to values equal to powers of two. This corresponds to noise source wordlengths of integral numbers of bits, and would most likely be convenient for ultimate microprocessor implementation anyway. Wordlengths from two to 10 bits have been catered for.

3.1.2.2 A Program to Configure the Noise Source to Produce A Specified Distribution: ALICUT

Once the target distribution has been specified, it is necessary to set up the flexible noise source so that its output PDF will tend towards this distribution. The setting up operation for Walker's technique is to compute so-called alias and cut-off values. The generation of the noise source output is then dependent upon the internal generation of two uniformly distributed and mutually uncorrelated pseudo-random sequences, denoted by U and X say. With each possible value of X is associated an alias value A(X) and a cut-off value C(X). An element in the noise source output sequence, Y_k, is generated as follows. A random pair (X_k, U_k) is drawn from the uniform distributions.

If $U_k \leq C(X_k)$ then the noise source output $Y_k = X_k$

If $U_k > C(X_k)$ then the noise source instead outputs alias of X_k, namely $Y_k = A(X_k)$.

The program ALICUT is an implementation of the algorithm to compute alias and cut-off values for the noise source, (Walker, 1974 and 1977). A notable new feature of the implementation written in the present work is that it uses integer arithmetic. The target distribution is entered as integer data, as previously noted. The algorithm uses integer arithmetic, and this obviates the numerical precision effects to which floating point versions are susceptible. The use of integer arithmetic also gives a substantial reduction in execution time: 35 minutes for floating point versus 10 minutes for integer, for a 512 point example.

The cut-off values, having been computed by integer arithmetic, are in normalised integer form. The use here of integer cut-off values does not introduce any errors in the operation of the noise source. This is because the random value to which the cut-off value is compared, is generated by an integer mechanism and has an inherent quantisation equal to that of the cut-off value.

The alias and cut-off values computed are stored in disk file RNGDATA.

3.1.2.3 A Program to Generate Pseudo-Random Sequences with Prescribed Distribution: RNGEN

The program RNGEN is a software model of a pseudo-random noise generator based on Walker's technique. Hardware versions of similar generators have been developed and interfaced to the Nova. However, the hardware versions are rigidly limited in the number of random variates which can be generated. Present hardware has an 8-bit output word, and can produce distributions of up to 256 variates. The speed of the hardware versions is of little importance here, because the digital filtering cannot be done in real-time with the facilities available. Also, a software generator is mandatory for simulation studies done via the IBM package. The operation of the software model is more readily appreciated because all the routines are written in Fortran, but in any case it has been desirable to keep the Nova and IBM packages as similar as possible.

It was felt necessary to provide the facility to generate distributions with large numbers of random variates. The main basis for this was the assumption that the PDF of the filtered signal could be adjusted by adjusting the PDF of the signal into the filter, as described in Section 2.3.2. The assumption is known to hold in the special theoretical cases of single frequency sinusoids and Gaussian noise, for linear networks. It has been found experimentally that the assumption does not hold for other cases, as is described in Section 4.1. Based on the above assumption, the design idea was to provide for very localised adjustment of the distribution generated for filtering, and this necessitated many variates.

A further factor motivating the investigation of test systems with many-variate noise sources, is the requirement that quantisation effects in the filtered stimulus be negligible with respect to those in the digital to analog converter producing the analog stimulus. This requirement arises when budgeting system errors to meet the test set back-to-back SNR specification. If virtually the whole system error power is allocated to the data converters, then presently realisable converter speed-resolution products are arrived at, as is shown in Chapter 4. As allowable error

power is allocated to (say) quantisation of the filtered stimulus, however, the converter requirements are pushed expensively closer to the present state-of-the-art specifications. Hence the send digital filter samples must appear to have been drawn from a continuum. It is clear for simple digital filters that the set of output sample values is constrained by the filter algorithm but also by the set of input sample values. While the quantitative importance of this effect can be expected to decrease rapidly with increasing filter order, it has been felt necessary to confirm this quantitatively as the test system specifications are severe. It will be seen in Chapter 4 that these effects are to an extent overshadowed by the experimental discoveries of Section 4.1.

A feature of this noise source model is that it is necessary to specify the number of noise samples required, as well as the value of a 'skip' parameter. The model has, of course, a characteristic pseudo-random sequence. The program always starts generation from the same point in this sequence. It is possible, by suitable specification of the skip parameter value, to apply to the subsequent filter a contiguous block of noise samples drawn from the characteristic sequence but starting at any point within the sequence. For a skip parameter value of N, the first N samples generated by the noise source are discarded, and are not presented to the filter.

Two uniform distribution noise sources are required within the model to generate the prescribed distribution, (Walker, 1974 and 1977). It was decided to program these uniform distribution sources in Fortran and to base them on the mixed congruential method, (Lewis, 1975, pp 5-6). The two uniformly distributed sources are incorporated as the subroutines RNGSUB1 and RNGSUB2 called by RNGEN. For reasons of ultimate hardware system implementation, it is desirable to have in the simulation noise source algorithms which are known and characterised precisely, and which can be repeated exactly on a variety of hardware systems. Hence it has been preferable not to draw upon computer vendor supplied assembly language routines. Linear feedback shift register (FSR) sources were considered, but these are not convenient to model in Fortran and in uncomplicated form show patches of high local correlation in the generated sequence (Hartley, 1975). It is felt that a FSR source would not be significantly easier to implement in a practical test system developed along the proposed lines.

It has been convenient to represent the outputs of the uniformly distributed noise sources as integer data. For the purpose of initial investigation, the noise samples are computed by modulo-(2**15) arithmetic,

and form a sequence of length 32768. It is noted that where the modulo-(2**m) remaindering is done by overflow of an m-bit register, there will be at best 2**m random variates. It is desirable to preserve the uniformity of this distribution when mapping the sub-generator output to select a final variate to be output or aliased. The mapping is facilitated if the number of variates in the target distribution is also equal to a power of two, and hence the constraint introduced in Section 3.1.2.1.

The use of short (15-bit) wordlengths in the uniform distribution sources makes it imperative that the constants be carefully selected. The aim has been to minimise, in some general sense, the auto-correlation of each of the generated sequences. Different algorithms have been used for the two uniform distribution sources in the hope that the cross-correlation of the two sequences will be insignificant. Selection of the algorithms has been approached on the following basis. Consider the sequence of samples X_i generated by the mixed congruential algorithm

$$X_i = A\,X_{i-1} + C \qquad (\text{mod } N)$$

Greenberger (1961, p 383) has shown that if C is set to zero to give a multiplicative congruential algorithm, then the maximum sequence length obtainable is N/4. In the present work this represents a stimulus period of only one second, assuming a noise source sampling frequency of about 8kHz, as motivated in Section 5.6. The intention has been to determine the required stimulus period by observing the asymptotic convergence of measured SQNR with increasing stimulus period. It was felt that a four second period would be just adequate; and hence the mixed congruential was selected over the multiplicative for this reason, rather than because of 'superstitions that these differ significantly in their statistical properties' (quote Coveyou and Mac Pherson, 1967, p 101).

Greenberger (1961, p 383) has shown that in order to ensure that the sequence has maximal length N, it is necessary only that :

for N equal to a power of two (32768 here),

C should be odd

and A = 1 (mod 4).

Within these definite requirements there is still considerable freedom of choice of constants. Greenberger (p 384) has suggested that we are still a 'considerable distance' from understanding how this choice may be best exercised to minimise, say, serial correlation of the generated sequence. He has suggested on intuitive grounds that A be of order $N^{\frac{1}{2}}$. Greenberger has derived a significant correction term to Coveyou's

approximate relation for the serial correlation coefficient with a lag of one sample period, but has assumed N to be on order (10**9). This formulation may be extended for other lags, but has not been used here as each analytic computation must be tackled as a new particular solution.

Coveyou and Mac Pherson (1967) have used Fourier techniques to predict the statistical performance of these sources, but have considered explicitly only much larger wordlengths. They have suggested that A should not be close to a simple rational multiple of $N^{\frac{1}{2}}$, and have cautioned against short wordlengths like that used on the IBM 360. They have also suggested that the multiplier A should not be chosen with few ones in the binary representation in order to facilitate shift-and-add implementations.

Graham (1977) in a more recent but less formal publication has given guidelines adopted in the present work. These are that :

A = 5 (mod 8),

and A should exhibit no simple pattern in the binary representation. C should be odd and near N*(1-1/√3)/2. The constants have been chosen to comply with these rules, and the statistical behaviour of the generators has been assumed to be adequate. Statistical tests have not been done to confirm this.

The sign bit of a sample leaving the overall noise source model has been generated by taking the unused most significant bit of the 16-bit word in subroutine RANDU. No detrimental effects have been observed experimentally.

3.1.2.4 The Digital Filters

The purpose of this section is to examine the send, noise and signal level digital filtering facilities which have been built into the simulation package. These filters are shown in Fig. 18 as the programs *FLTS*, *FLTRN*, and *FLTRS* respectively. It is emphasised that the filter schemes to be discussed are in a form suited to initial experimental determination of tester design requirements, rather than having pretensions to suitability in a practical test system design.

The filter requirements of the simulator may be very varied. The band-edge frequencies of any of the filters may need to be set arbitrarily. Filters with widely different orders (numbers of poles) may be required. In order to establish what order of filter could be expected in typical simulations, it was necessary to perform an approximate analysis of a representative test situation. This analysis has been done with the assumption of filter amplitude responses which are stepped, so that a

filter may be characterised by its band-edge frequencies and an average lower and upper stop-band attenuation. This tedious analysis will not be detailed because the simulator is now available as a much more precise analytic tool. However, an important result of this analysis is that average stop-band attenuations of over 80dB were to be anticipated. This is a severe requirement, and arises mainly because of the assumed 12-bit data converte. resolutions. It has thus been decided to provide in the simulator filter ... with up to 40 poles, but this may be readily extended if required.

The simulator system design idea has been to allow the operator to easily 'plug-in' a filter module of the order he requires, and to specify its band-edge frequencies. Because many different digital filters may be required in simulations, it has been necessary to incorporate in the simulation package programs to automatically design and synthesise those filters. The key to the implementation of this scheme of programmed design and realisation is the recognition that all the necessary filtering may be performed by bandpass filters. Some steps in the provision of a filter are outlined in Figure 20.

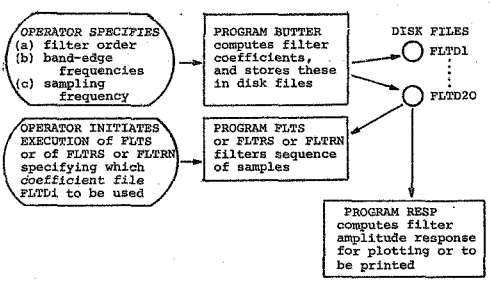


Figure 20 Design and Implementation of Bandpass Digital Filters

It has been felt unnecessary to limit the digital filters in the test system to non-recursive types. The use of recursive types has, of course, the advantage that the zeros allow a desired frequency response to be obtained with fewer operations than would be necessary if only poles were used (Gold and Rader, 1967). A non-linear phase response is of no consequence in the cases of the receive side filters FLTRN and FLTRS, because only the powers of the filtered signals are computed. It has been felt that a send filter with linear phase is unlikely to be significantly advantageous in generating a test stimulus with speech-like PDF. This opinion has not been proved by formal analysis, however. The non-availability of analytic techniques to handle this general problem is discussed in Section 5.1.

It may be desired to filter a signal sequence by the repeated operation of a filtering algorithm. This is readily feasible within the present structure because filter input and output sequences are stored in separate disk files. An implication is that it is reasonable, although possibly cautious, to avoid filters with ripple of the amplitude response within the passband. Consequently, and to start a library of filter algorithms, the program to compute filter coefficients gives only Butter-worth type filter responses. The operation of this program, BUTTER, is outlined in Figure 21.

The programs to design and implement the digital filters were written by the present author specifically for the simulators. These programs are, however, distinct modules which are readily usable for general purposes outside of the present work. Very little knowledge of digital filtering details is required in order to specify and implement filtering using these programs.

The lowpass-to-bandpass transformation for analog filters is well-known. The bi-linear transformation is also a well-established technique. The use of the bi-linear transform design technique may be seen as almost mandatory in the present application, because the noise measuring filter must be definitely broadband (Kaiser, 1972). All the filters to be designed are required to have flat response within the passband. Hence the frequency 'warping' aspect of the bi-linear transformation requires only that the prototype analog cut-off frequencies be pre-warped in compensatory fashion, but the effect is otherwise an advantage here. No further attention has been paid to aliasing effects.

It has been felt unnecessary to limit the digital filters in the test system to non-recursive types. The use of recursive types has, of course, the advantage that the zeros allow a desired frequency response to be obtained with fewer operations than would be necessary if only poles were used (Gold and Rader, 1967). A non-linear phase response is of no consequence in the cases of the receive side filters FLTRN and FLTRS, because only the powers of the filtered signals are computed. It has been felt that a send filter with linear phase is unlikely to be significantly advantageous in generating a test stimulus with speech-like PDF. This opinion has not been proven by formal analysis, however. The non-availability of analytic techniques to handle this general problem is discussed in Section 5.1.

It may be desired to filter a signal sequence by the repeated operation of a filtering algorithm. This is readily feasible within the present structure because filter input and output sequences are stored in separate disk files. An implication is that it is reasonable, although possibly cautious, to avoid filters with ripple of the amplitude response within the passband. Consequently, and to start a library of filter algorithms, the program to compute filter coefficients gives only Butterworth type filter responses. The operation of this program, BUTTER, is outlined in Figure 21.

The programs to design and implement the digital filters were written by the present author specifically for the simulators. These programs are, however, distinct modules which are readily usable for general purposes outside of the present work. Very little knowledge of digital filtering details is required in order to specify and implement filtering using these programs.

The lowpass-to-bandpass transformation for analog filters is well-known. The bi-linear transformation is also a well-established technique. The use of the bi-linear transform design technique may be seen as almost mandatory in the present application, because the noise measuring filter must be definitely broadband (Kaiser, 1972). All the filters to be designed are required to have flat response within the passband. Hence the frequency 'warping' aspect of the bi-linear transformation requires only that the prototype analog cut-off frequencies be pre-warped in compensatory fashion, but the effect is otherwise an advantage here. No further attention has been paid to aliasing effects.

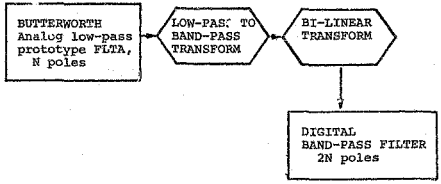


Figure 21 Filter Coefficient Calculation in Program BUTTER

The lowpass-to-bandpass transformation and the bi-linear transformation have been implemented in BUTTER as a single set of calculations. This is based on the analytic formulations of Stanley (1975, pp 194-198), with the extensions detailed in Appendix F. However, the operation of BUTTER must be clarified by the illustration of Figure 22.

The effect of the combined transform on an arbitrary complex conjugate pole pair has been formulated algebraically and is detailed in Appendix F. The validity of the decomposition and transformation scheme is intuitively obvious, but no formal proof can be cited to show $H_T = H_1.H_2.H_3 \ldots$ The program RESP may be run to calculate the frequency response of any filter designed via BUTTER, and this response may also be plotted (with ordinate in decibel form) as required.

The programs FLTS, FLTRN, and FLTRS may be used to implement any filter designed using BUTTER. The form of implementation is the cascaded section form shown in Figure 22. Naturally, each of the cascaded sections has only four poles. Each section has been realised directly from the impulse transfer function. Various other realisation forms are given in the literature, but the form used appears to suffice here.

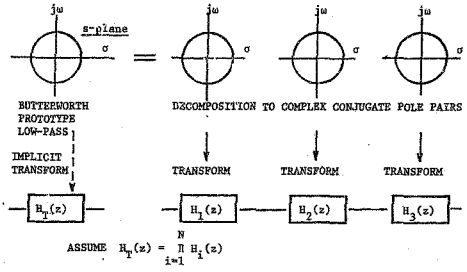


Figure 22 Transform Procedure using Pole Pairs in Program BUTTER

Double precision arithmetic has been used throughout all the filter programs. This gives a 64-bit representation of a floating point datum. The simulator itself may be used to readily demonstrate that single precision arithmetic is inadequate here.

It may be expected that simulations will involve precise power measurements and fairly high order filters (40 poles is not unreasonable for the send filter). Consequently, account has been taken of the step response times of the filters; all signals have been assumed to be periodic with period equal to that of the test stimulus. When a block of signal samples is to be filtered, the last (in time) 1000 samples of the block are first passed through the filter to set the filter 'initial conditions' prior to filtering the block of samples itself. The adequacy of this 1000 sample 'tail' in allowing filter transients to decay has been confirmed by repeating representative simulations of Section 4.2 using an 8000 sample tail. The SQNR measurements are not changed by this longer tail.

The digital filtering modules included in the simulation package are very easy for the non-specialist to use, and may thus be of wider interest than test system design. For more refined test system design, however, an

inverted (type II) Chebyshev prototype would probably be preferable to the Butterworth. Operator specification of the Butterworth lowpass prototype order is presently limited to even order only, but this has not been a significant limitation.

In simulations it may be required to plot the probability distribution of the signal before and after filtering. It has been convenient to provide only the facility to plot the probability distribution of the filtered signal, this being of greater interest. The probability distribution of the pseudo-random noise prior to filtering may be plotted via the same mechanism by using the all-pass 'filter' FILTO.

3.1.2.5 Programs to Simulate a Linear Digital to Analog Converter (QUANTO), Companding PCM Encoder (IDLPCM) and a Companding Decoder (CONVERT)

The program QUANTO performs several important functions; these are the absolute numerical scaling and quantisation of the test sequence, the accumulation of probability distribution values of the filtered stimulus, and the generation of disk files to test the PCM system and to plot the probability distribution. This section seeks to outline the aims of these operations, and how they relate to the programs IDLPCM and CONVERT.

The program QUANTO reads the output of the send filter from file FLTO. These are single precision real data. The data must scaled and quantised to give the particular required stimulus level with the required D/A converter resolution. The D/A may be a unit actually interfaced to the Nova, such as the 12-bit unit described in Appendix B, and used with a hardware PCM system. Alternatively, purely software simulations require the quantisation effect of a D/A converter to be simulated, and this may also be done by QUANTO. These latter simulations are performed with a software simulation of a PCM codec, as implemented in the programs IDLPCM and CONVERT. The idea has been to perform all the scaling operations in real arithmetic, and to quantise the D/A converter signal to the required integers only at the final stage.

For a specified D/A converter resolution, the peak stimulus value may be calculated. The operator may specify how far down on this peak value the test stimulus is to be. Variations of this specification are equivalent to using a 'digital' attenuator, as the attenuation operates before the D/A converter. In order to allow investigation of the trade-off between digital and analog attenuation and D/A converter resolution, the program IDLPCM allows an equivalent analog attenuator to be set. For the

rest, IDLPCM is a straight-forward software model of a ramp type encoder based on the COMDAC companding D/A converter used in the hardware PCM system.

The COMDAC companding D/A used in the hardware PCM system and modelled in IDLPCM, implements a segmented approximation to the μ-255 companding law. This implementation has been available as the DAC-76 integrated circuit since the outset of the present work. While an A-law compander is anticipated in telephone PCM systems in South Africa, the μ-law hardware provides a preferred basis for more general instrumentation (Hanrahan, 1978). Differences between A-law and μ-law SQNR characteristics are not important in the present work.

Some numerical scaling is required to match the QUANTO D/A converter output range into the IDLPCM simulated encoder input range. This is done using type real arithmetic. The output of the simulated encoder is in the same binary form as that of the actual PCM system and is written to the same disk file called DATA.

After the stimulus sequence has been scaled to give the required RMS level, the program QUANTO quantises the sequence to integer values. It is then necessary to explicitly clip the signal peaks to the extreme values of the D/A converter range, as overflow beyond the D/A converter wordlength may result in incorrect operation. In the case where the hardware D/A converter is to be used, the bipolar integer signal is level shifted to be within the range 0 to 4095, as the offset is subtracted in the analog circuitry. The output sequence is then written to the disk file TEST, and this will differ in form depending on whether a hardware of software PCM encoder is to be used. The probability distribution of the output sequence is written to the disk file PLOT and may be plotted from there as required. The form which TEST has for a hardware PCM encoder, and that which PLOT has for application to the plotter D/A converters, is clarified in the next section. The operator may specify the number of points required in the probability distribution, and this must not be greater than the number of output levels possible within the specified stimulus D/A conversion.

The program CONVERT is a model of a PCM decoder based on the COMDAC companding D/A converter used in the hardware. The use of the software decoder is recommended for quantisation noise studies, whether the hardware or software encoder is used, unless decoder imperfection effects are specifically required. An ideal decoding formula has been published for the COMDAC, and this has simply been used.

For more accurate software simulations with the IBM package, a facility has been created to model the linear A/D converter separately, as opposed to lumping its noise contribution with that of the linear D/A converter. The linear A/D converter and its preceding analog attenuator are modelled in the program ADCON. This facility does not exist in the Nova package, because Nova based studies would sensibly use the 12-bit hardware A/D converter.

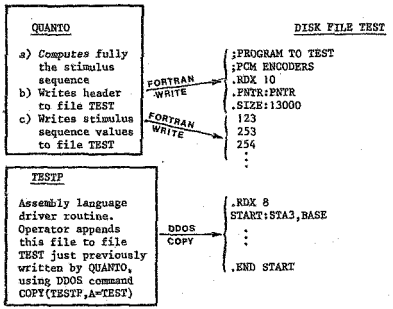
3.2 Software - Hardware Interaction

3.2.1 The Fortran/Hardware Interface

The aim has been to perform the signal processing off-line and by high level language programs wherever there is no compelling real-time requirement. Where hardware tests are to be done, however, it is essential to generate the analog test stimulus in real-time. Although not an essential requirement, it has been convenient to record the quantised output of the PCM system in real-time, i.e. within one stimulus period. The various peripherals have hardware interfaces to the Nova (as detailed in Appendices A, B, C) and these are controlled by assembly language routines. The interaction between the hardware interface and the assembly language routines may only be seen in the detailed terms of Appendices A-D. The purpose of this section is to establish the ideas which led to the design of the Fortran/assembly language interface.

It is required here to pass data from a Fortran program to an assembly language routine, and then to pass data from an assembly language routine to a Fortran program. These data correspond to variable values within the Fortran program. A conventional procedure exists to effect these transfers, and this is described in the DOS 1.6 Programmer's Reference Manual (1974). The conventional procedure is not suitable for applications such as the present one, as is detailed later. An alternative scheme has been devised to meet the present requirements.

Let us consider the generation of the test stimulus. The values of the test stimulus sequence have been computed at the end of the program QUANTO. It is required that the D/A converter assembly language driver routine should output this sequence in real-time. While it is required that the driver routine should run in real-time, it is emphasised that it is not necessary that the Fortran to driver data transfer be done in real-time. It is not desirable to incorporate in the driver routine computational procedures to be performed on the stimulus sequence. The scheme implemented is outlined in Figure 23.



Disk file TEST is built up as shown above. The file TEST is then regarded as a normal assembly language program, and is assembled and run via DDOS commands.

Figure 23 Transfer of a Data Block from FORTRAN to Assembly Driver.

It is felt that the scheme above addresses the present requirements better than the more conventional (DGC) procedure. A major advantage of this scheme is that it allows the longest possible test sequences to be used. The Fortran program is not required at the driver run time, and occupies no core here.

Also, the scheme is more compatible with the simulation package structure in that it is development orientated. Hardware development is likely to involve the creation of new driver routines. The development of a new driver is very greatly facilitated in that is requires only familiarity with Nova assembly language programming. The conventional approach demands a detailed understanding of the internal Fortran array handling subroutines and of the calling and receiving sequences. These procedures are complicated here by the inescapable need to pass the data as an array rather than a list of variables, and are detailed in the 'Fortran Run Time User's Manual 093-000068' (1975). The proposed scheme is so simple that debugging the driver would require virtually no attention to be paid to the Fortran/driver interface.

Furthermore, the addition of a new driver does not necessitate recompilation, assembly and linking of the Fortran program, as would be required by the conventional approach. When a range of drivers has been built up, execution of the required driver may be selectively initiated in a similar manner to that used for other programs in the package.

The necessity to assemble and link the driver each time for new sequence data might possibly be seen as a disadvantage of the proposed scheme. If this is felt to be significant, the sequence data may be written to the disk file in binary form so that these may be superimposed on an 'old' driver already in core image format. This involves loading first the 'old' driver program, then the sequence data. Assembly of the driver would only be necessary if the sequence length were changed. Such a change would involve several hours of processing and the driver re-assembly time would be insignificant. The need for the Fortran program to write the test sequence to a disk file cannot be avoided for reasonable length sequences, and is independent of the Fortran/assembler interface used. This arisss because QUANTO must accept the sequence as type real data, and must write the processed signal to some or other disk file as only 16 kilowords of core store are available. Hence if the conventional Fortran/assembler interface were used, it could not be incorporated in QUANTO.

The proposed interface scheme has been used also to pass data to the X-Y plotter driver. It has often been advantageous to keep on a separate user disk the linked versions of particular driver programs and data. Hence if a particular plot is required again, for comparison

purposes or to a different scale, the linked version if simply run again. Similarly, the various test sequences developed are kept in this compact but complete form, and no other disk files on the send side need be retained. It is almost mandatory to retain the various test sequences developed, because each represents substantial computation.

3.2.2 Assembly Language Programs

The aim of this section is to outline the operation of two representative assembly language callable driver routines. The routines are called TESTP and PLOTP, and extensively commented listings of these are given in Appendix D, together with the necessary hardware details in Appendices A, B, C. The routine TESTP applies a test stimulus to the 12-bit D/A converter, and stores the quantised output of the PCM encoder A. The routine PLOTP sends co-ordinate pair values to the dual 10-bit D/A converters, and may be used to plot probability distributions on the H-P 7005B analog X-Y plotter.

On receiving program control from DDOS, TESTP must first store the DDOS system table base address passed to it for later use. The situation then is that the sample values to be output are stored in a contiguous block of core. TESTP sets the PCM system mode required. When TESTP has synchronised to the PCM system sampling clock, it begins to output the test stimulus and input the encoded signal. These data transfers are done in synchronism to the PCM system sampling clock throughout. The initial synchronism is established precisely by checking for a change in the PCM system BUSY and DONE flags rather than the level of one of these flags.

The operation is as follows. A stimulus sample is loaded into accumulator ACC 1 from a location in the contiguous block. The sample is then output to the D/A converter, where the analog output is allowed to settle. Some time later, the PCM system starts encoding the sample from the D/A converter. When this encoding conversion is complete, TESTP inputs the encoded sample. The encoded sample is then stored in the core location from which the stimulus sample value was taken at the beginning of this operation. The operation is repeated at the PCM sampling rate, for consecutive locations in the contiguous block of samples, until every stimulus sample in the block has been replaced by its encoded version.

The block of encoded samples is then transferred from core to a disk file called DATA. This transfer is simply effected by the driver calling the primitive function O.IMG which is part of DDOS. Program control is then returned to DDOS via a call to the DDOS primitive function D.HOM. These primitive functions are loaded by a DDOS driver, but the system table base address is required for the calls. The procedures for calling primitive functions are detailed in the DDOS 1.6 Programmer's Reference Manual 86-0140050-4 (1974). It is noted that TESTP assumes that the DDOS core resident area has not been overwritten when TESP is loaded. Hence test sequence length must be kept below some 14 000 samples for hardware tests.

The routine PLOTP is based on the same programming ideas as TESTP. Only the Y-values are passed to the routine, and these are plotted at equidistant abscissa values. A simple scaling scheme in QUANTO calculates the required abscissa increment so the number of points to be plotted may be varied. The routine PLOTP finally plots points on the X and Y axes indicating one tenth divisions of the ranges.

3.3 A Summary Description of Test System Software

In Chapter 2 the need for a test system simulator and some test set hardware was established. In Chapter 3, the more detailed requirements of each functional block of the simulator have been established. The requirements for each functional block have been considered in the context of the computing environment. These considerations have defined composite requirements which each software block has been designed to meet. Descriptions have been given of a simulation package structure which meets the present needs, and the program module designs have been described.

It has been shown that two simulation packages are useful in the present work. Both packages are written in Fortran, but they are designed to execute on different computers. The Nova package provides the basis for future test set hardware studies. The IBM 370 based package allows test set design principles to be rapidly established, and has been used for the simulation studies described in Chapter 4.

The software described in Chapter 3 has given rise to some products which may be of interest outside of QD measurement. The file program network structure of the Nova package exemplifies a solution to the general problem of accommodating large programs or data arrays on a relatively

modest computer. The digital filtering programs are flexible and easy to use. The Fortran/assembler interface is novel, and is an attractive solution to a class of commonly occuring problems not addressed by the conventional Fortran/assembler interface.

The use of the simulator to quantify some fundamental ideas of QD measurement is reported in Chapter 4. The simulator has been found particularly valuable in that its use has resulted in the experimental discovery of a most important fundamental aspect of QD measurement by non-Gaussian stimuli. The discovery is reported in Chapter 4, and its significance is discussed in Chapter 5.

4 TEST SYSTEM SIMULATIONS PERFORMED

The purpose of this chapter is to outline some initial simulations performed using the simulator described in Chapter 3. These initial simulations are intended to demonstrate the ease with which precise quantitative results may be obtained using the simulator, and hence to suggest the usefulness of the simulator in further work in this area. No attempt has been made to exhaustively exploit the potential of the simulator.

Many of the initial simulations have been done using the IBM package. Most PDF's have been plotted on a hard-copy terminal with software based on the GRAPH(LIB) program extant on the WITS system. Some plot scaling and formatting routines have been written into the IBM simulation package, so only small modifications have had to be made to GRAPH.

4.1 Probability Distribution Modelling

Experiments were conducted to establish the random source target PDF required to give a speech-like PDF for the filtered signal. Let these PDF's be denoted as P1 (for the signal prior to the send filter) and P2 (for the filtered signal). A widely differing variety of distributions has been used for P1. The shapes of some of these distributions are sketched in Fig. 24.

The target distribution and the distribution achieved by the random source have been plotted. This has graphically verified that the random source achieves a distribution very close to the target, with a sequence length of the order of 32 000. Hence the shapes in Fig. 24 are quite accurately representative of the distributions actually input to the send filter.

It is desirable to concentrate filtering effort at the send filter, as is demonstrated in Section 4.6. A 40-pole send filter has been used here and for all experiments other than those of Section 4.6. In isolating the effects of imposing practical limitations on various sections of the test set, it is often convenient to be able to regard specific filters as being

ideal rectangular types. Where it has been required to have such ideal filters in a simulation, then those particular filters have merely been specified as 40-pole types. This convenient round number implies considerable computation, but safely removes finite filter roll-off effects from the simulation. This assurance is based on a comparison of some simulation results using first 32-pole filters and then 40-pole filters (Section 4.6.2.)

It has been found that each and every P1 shown in Fig. 24 gives a filtered PDF, P2, that is approximately Gaussian in shape. It appears from the plotted PDF's that the form of P2 is not noticeably a function of P1. As the filter is linear, it is anticipated that P2 would be Gaussian for the case where P1 is Gaussian. Hence P2 will be referred to as Gaussian for any P1, although it has not been analytically proved that P2 tends to a strictly Gaussian form in general.

No specific reference to the above effect has been found in the literature. Immediate questions arise as to whether the tendency of the filter output towards a Gaussian PDF is avoidable by specific minor changes in the test set. Could this effect be avoided by, for example, using non-recursive filters with linear phase, or by using filter prototypes other than the Butterworth? A general analysis is required which rigorously defines the sufficient conditions for convergence to the 'Gaussian' output. The feasibility of this analysis and its significance is considered in Chapter 5. Suffice it for the present to note that there exists, in all the particular situations simulated, an effect reminiscent of the central limit theorems. It has consequently not been possible to compare SQNR's measured using Gaussian stimulus PDF to those measured using more speech-like PDF's. In Chapter 5, an algorithm is proposed which circumvents the limitation to Gaussian stimuli.

The arbitrary spike PDF shown in Fig. 24 was introduced in the hope of inducing the Gaussian P2 to peak more at low amplitudes. This is unsuccessful even when the filter bandwidth is increased from 100 Hz to 1500 Hz. It appears that the filter bandwidth does not significantly influence the form of P2.

Some sample plot printouts are given in Appendix G.2. The practice has been to print the data file corresponding to the particular simulation below the PDF plot; this fully details the simulation and automatically ensures that there is no confusion when many simulations are performed.

4.2 The COMDAC SQNR versus Test Stimulus Level Characteristic --Measurement of a Reference Curve

The purpose of this group of simulations was to establish the 'actual' quantisation noise characteristic of the ideal COMDAC model. This characteristic was measured by specification of an ideal tester with the following parameters :

sequence length	32 000 samples
send filter	450 - 550 Hz
noise filter	860 - 3400 Hz
filter order	40 poles (all filters)
D/A resolution	30 bits
A/D resolution	30 bits
random source target PDF	Gaussian
system sampling frequency	8 kHz

This tester is ideal in that all the filters are 40-pole types, and both the data converters are 30 bit types. The convenient round number of 30 bit resolution serves to remove from the simulation the effects of finite data converter resolution. The simulator has been used to provide assurance that 30 bit resolution is sufficient for this purpose, and it has been found that data converter quantisation noise has a detectable effect on SQNR only when resolution is reduced to 16 bits (Section 4.6.1).

The precise numerical results are given in Appendix G.3, and are shown in Fig. 25. These results are not directly comparable with the published COMDAC performance curve (Precision Monolithics Inc., 1976) because the PMI curve has been established by sine wave tests. The results shown in Fig. 25 will be used as the reference 'actual' performance goal for evaluating less precise test sets. They are useful, also, in the measurement of the encode and decode noise components introduced in an imperfect hardware codec in addition to the quantisation noise.

4.3 The Variation of Measured SQNR as a Function of Test Length

A group of simulations has been done to quantify the variation of measured SQNR of a typical codec as a function of the test sequence length. The COMDAC software model has been taken as representative. Fixed parameters of the test set were specified at ideal reference levels:

send filter	450 - 550 Hz
noise filter	860 - 3 400 Hz
filter order	40 poles (all filters)

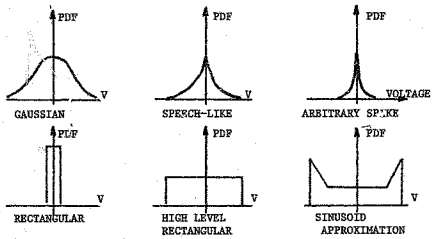


Figure 24 Variety of Random Source Target Distribution Shapes Used.

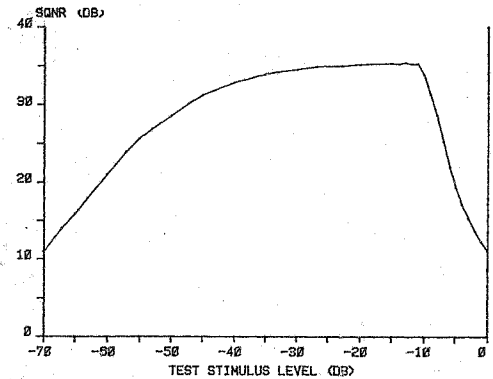


FIGURE 25 THE SQNR VERSUS TEST STIMULUS LEVEL CHARACTERISTIC OF THE COMDAC.

D/A resolution	30 bits
A/D resolution	30 bits
random source target PDF	Gaussian
system sampling frequency	8 kHz

The numerical results are given in Appendix G.4 and are illustrated in Fig. 26. As anticipated, the measured SQNR settles to a level with increasing test length. The purpose here has been to determine the minimum test length which gives measurements within a specified band around the settling level. This minimum acceptable test length is an important factor determining the economics of both test set software simulation and hardware design.

The variation of measured SQNR with the test length is also a function of the test stimulus level. Consequently, three representative stimulus levels have been chosen as parameters of the simulations. Referring to Fig. 25, levels of -7dB, -13dB, and -43dB were selected as representing the different encoder loadings. These levels have been set by specifying the digital attenuator setting. The actual stimulus level may appear to differ slightly from the digital attenuator setting when high stimulus levels are being used. This occurs as a result of peak clipping in the D/A. The program QUANTO informs the user of the level of the peak-clipped stimulus, and this is typically 7,2dB down when the digital attenuator has been set for 7,0dB.

This is intended only to indicate the peak clipping power. The effect is not an error source in ideal test set simulations. When the send side analog attenuator is set to 0dB, then the D/A converter peak range maps exactly into the PCM encoder peak input range. Hence the stimulus level in the above example should be taken as 7,0dB exactly, and the peak clipping notionally attributed to the PCM encoder.

The simulations of this section were unintentionally performed with the send side analog attenuator set to give a gain of 0,135dB (i.e. S =-0,135). This has no significance in terms of the purpose of this section, but does explain why there is not precise numerical coincidence between this section and the previous section at the 32 000 sample sequence length point.

The scales of the graphs in Fig. 26 are not all the same as the graph varies much more for high stimulus levels (-7dB). The simulator noise source has a period of 32767 samples. Short length stimuli are periodic

repetitions of segments of this characteristic period. In the present group of simulations, the various length stimuli are all taken starting from the same point in the characteristic sequence. It is expected that the precise form of the graphs (Fig. 26) would depend upon the starting point chosen. This may be investigated by repeating the present group of simulations using a different value for the ISKIP parameter, but has not yet been done. The shape of the graphs might also be dependent upon the particular characteristic sequence of the source modelled. It would be interesting to investigate the effects of various new sets of constants in the random generator algorithm, and so on.

4.4 Measured SQNR as a Function of Test Stimulus Frequency Domain Parameters

4.4.1 Measured SQNR Variation with Stimulus Centre Frequency

In this group of simulations, the SQNR of the COMDAC model has been measured using a 100 Hz wide band of noise. Measurements were taken with the stimulus band centred at various frequencies. Fixed parameters of the test set have been given ideal values. Particularly, the noise measuring filter passband has been fixed at 860 - 3 400 Hz as is conventional.

The numerical data are given in Appendix G.5.1, and the results illustrated in Fig. 27. Clearly the error spectrum is not injection frequency independent, and, again, large variations of measured SQNR occur at high stimulus levels.

4.4.2 Measured SQNR Variation with Stimulus Bandwidth

These simulations were performed with the stimulus centre frequency at 500 Hz. The stimulus level measuring filter on the receive side was held equal to the send filter for the bandwidths simulated. As narrow bandwidths were to be investigated, the filters were all initialised by an 8000 sample 'tail', as opposed to the usual 1000 samples. Comparisons of results with the different filter initialisations showed only minor differences.

The measured SQNR is virtually independent of the stimulus bandwidth at moderate and low stimulus levels. This is perhaps surprising, but is probably due to the very long stimulus length and averaging time used here (32000 samples, 4 seconds).

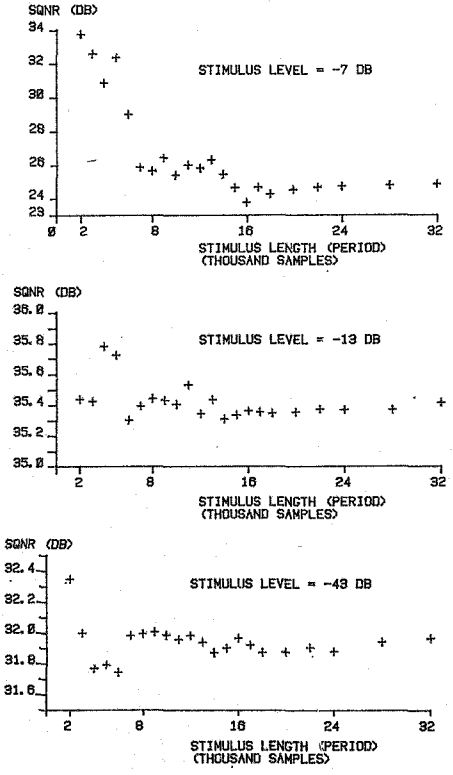
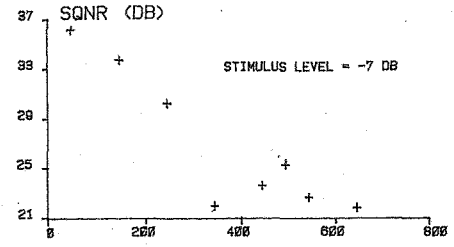
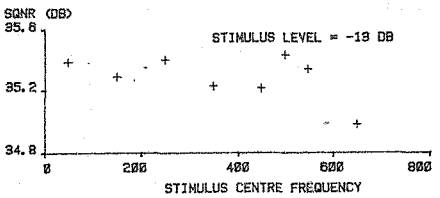


FIGURE 26 VARIATION OF MEASURED SQNR WITH TEST SEQUENCE PERIOD.





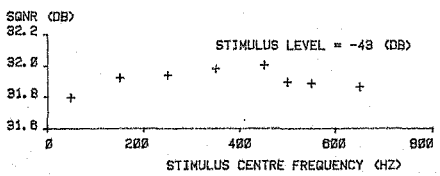
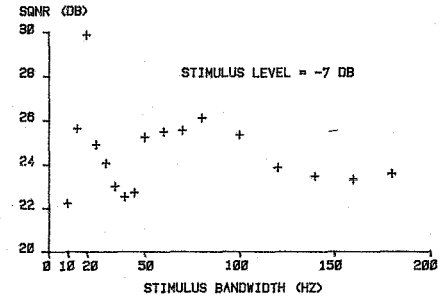
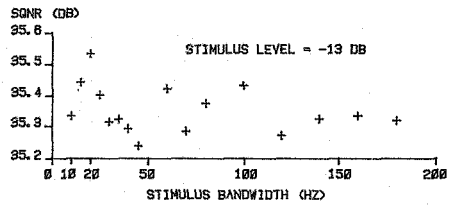


FIGURE 27 VARIATION OF SQNR MEASURED AS A FUNCTION OF NOISE BAND CENTRE FREQUENCY.





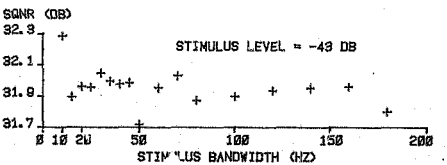


FIGURE 28 MEASURED SQNR AS A FUNCTION OF STIMULUS BANDWIDTH.

At high stimulus levels (-7dB), the SQNR measured is very strongly influenced by the stimulus bandwidth. Stimulus bandwidths of 100 Hz and 180 Hz give SQNR values differing by more than 1dB. This reinforces the impression that the achieveable accuracy in SQNR measurement will be reduced at high stimulus levels.

4.5 Determination of Data Converter Signal Level to Optimise Test Set Performance

The simulations of this and the subsequent sections are aimed at finding indications of likely major parameter values in a practical hardware test set design. The conventional measurement technique has been assumed for the present purposes, together with an implementation scheme consistent with the proposals of Section 2.5. The aim now is to introduce test set data converters of limited resolution, and filters of lower order, and to establish the minimum resolutions and orders acceptable.

Initial attention has been focused upon the data converters. The purpose of this section is to establish the attenuator configuration which optimises data converter performance. The relevant area of the test set and some notation are shown in Fig. 29.

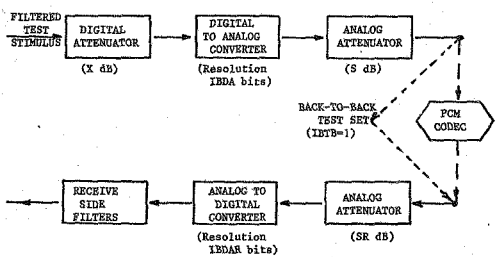


Figure 29 Test Set Data Converter and Attenuator Sections.

In a design environment where data converters are costly, no use will be made of digital attenuation to simplify the analog attenuator requirements at the send side. Initial examination will be of such an environment, and it is sought to establish the value of X (dB) required to minimise the IBDA and IBDAR resolutions necessary. This has been approached in the following manner. Simulations have been done where :

IBTB = 1 (test set back-to-back)

S = 0 dB

SR = 0 dB

IBDA = 12 bits

IBDAR = 12 bits

X(dB) has been varied, and the test set SNR measured. Other test set parameters have been set to reference ideal values.

Numerical data are given in Appendix G.6 and the results are illustrated in Fig. 30. While arbitrary values of IBDA and IBDAR might have been chosen, the 12 bit situation selected will be of interest in the light of the results of Section 4.6.2. Also, 12 bit hardware data converters have been incorporated in the hardware test facility (Appendix B). As noted previously, a test set SNR of 63dB is required to measure a SQNR of 43dB to within 0,1 dB. The 12 bit test set can have an adequate SNR of 62,5dB when X = 11,6dB and S = SR = 0 dB.

It is apparent from the approach shown in Fig. 29 that optimal use of the test set will be the following. X should be fixed at some point close to the optimal point given by the test set SNR criterion. Further simulations have accepted this criterion, and have fixed X = 11,6dB. The PCM stimulus level is then varied by S, which is a negative quantity in dB's when a gain is required for stimulus levels above -11,6dB. In order to maintain test set SNR independent of stimulus level, it is desirable to have tracking so that SR = -S dB. A unity gain PCM may be conveniently assumed.

4.6 Investigation of Test Set Design Trade-Offs between Data Converter Resolutions and Filter Orders

The aim in this section has been to simulate measurements as would be made by practicable test sets on the COMDAC model, and to compare these measurements with those of the 'ideal' test set of Section 4.2. Differences are called 'errors', and are seen to arise due to reduced data converter resolutions and filter orders.

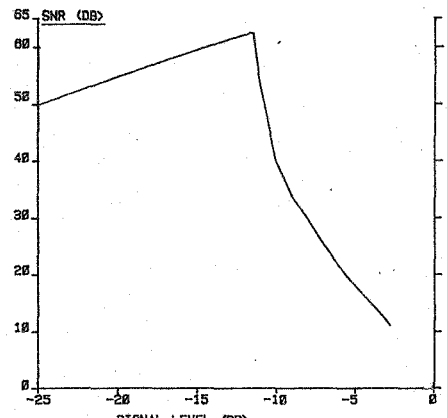


FIGURE 30 TEST SET SNR AS A FUNCTION OF DIGITAL ATTENUATION.

All simulations have assumed an equivalent send filter order of 40. This is because the send filtering for a hardware test set is not seen as a problem. The test stimulus may be held in read-only memory (ROM) or may be generated by a mechanism requiring neither noise source nor send filtering. One such mechanism is proposed in Chapter 5.

4.6.1 Effects of Constrained Data Converter Resolutions

In a test set along the lines proposed in Section 2.5, the data converters, of course, add their own quantisation noise power to that of the PCM system. Simulations have been performed to establish the quantitative significance of these additions, when measurements are made on a PCM system like the COMDAC.

The simulations of this section examine the effects of constraining the data converter resolutions, but have retained high filter orders (40 poles) for the receive side filters. The results are shown in Table 1, and numerical details are given in Appendix G.7.1. These results are of practical interest in a design environment where digital filtering capabilities are not hard pressed, and indicate the minimum data converter requirements. It will be seen in the next section that these ideal filters differ from the minimum acceptable filters by a factor of less than two in terms of computational effort. It is noted that all the simulations of this section and the next, use the fixed digital attenuator setting of 11,6dB derived in Section 4.5.

The simulations resulting in Table 1 have generally been combinations where the digital to analog (D/A) converter resolution is greater than, or equal to, the analog to digital (A/D) converter resolution. This would minimise the data converter engineering effort because D/A converters are simpler to implement than A/D converters, for the order of resolution required here. However, if the test stimulus is to be stored in ROM, the system cost minimisation differs. A reduction of D/A converter resolution by N bits would save N.L bits of storage for a test sequence of L samples, as well as a slightly reduced D/A cost. However, the cost of A/D converters increases rapidly for converter resolutions of more than 12 or 14 bits for the conversion speed required (125 μs for a telephone application). Further discussion of possible hardware test set architectures is deferred to Chapter 5, merely noting that Table 1 is not intended to be comprehensive.

4.6.2 Filter Requirements with Practicable Data Converter Resolutions

The simulations of this section take realisable data converter resolutions, and seek to establish the minimum filter orders required in conjunction with these converters. Measurements are again done on the COMDAC model, and these give errors with respect to the ideal test set of Section 4.2. The purpose is illustrative only, because there are more suitable filter types than the Butterworth prototypes used.

In order to avoid complicating the interpretation of the results, the order of the receive side stimulus level measuring filter has been held equal to the order of the noise measuring filter. This implies that the signal level filter is overspecified. Further simulations might usefully be done to determine the optimal proportioning of filtering effort between signal level and noise measuring filters. However, the assumption of equal receive filter orders is a useful starting point. If the receive filters are implemented by one microprocessor each, for example, the order of the stimulus level filter may be reduced from the 'equal order' specification. The excess capability of that microprocessor might then be used to control the simultaneous test stimulus generation.

The simulation results are shown in Table 2, and numerical details are given in Appendix G.7.2. A test set with a 14-bit D/A converter and a 12-bit A/D converter, for example, has been denoted in Table 2 as a '14/12-bit set'.

It is interesting to note particularly the Table 2 entries relating to sets with receive filters of 10 pole pairs per filter. It is clear that, over the range of data converters shown, test set performance is a stronger function of filter order than of data converter resolution. As evident for the 16/14-bit set, there is an abrupt limit to the minimum filter order acceptable. It appears that a satisfactory test set could be based on 12-bit data converters, with 24 pole receive filters. The receive filter order required is expected to drop if (say) inverted Chebyshev prototypes were used instead of the Butterworth. Comparatively minor additions to the simulation package would allow the extent of this drop to be determined.

TABLE 1 Errors in SQNR Measured w: a Practicable Data Converters and High Order Filters (All errors in dB)

	8 bit A/D			10-bit A/D		
D/A RESOLUTION	10	12	14	10	12	14
ERROR at - 7dB	0,163	0,156	0,160	0,087	0,077	0,080
ERROR at - 13dB	1,537	1,508	1,491	0,163	0,146	0,135
ERROR at - 43 3	0,766	0,762	0,740	0,076	0,062	0,048

	12-bit A/D			14-bit A/D		
D/A RESOLUTION	12	14	16	12	14	16
ERROR at - 7dB	0,009	0,012	0,012	0,013	0,017	0,017
ERROR at - 13dB	0,035	0,030	0,021	0,029	0,022	0,012
ERROR at - 43dB	0,013	0,001	0,001	0 015	0,001	0,001

TABLE 2 Errors in SQNR Measured with Practicable Data Converter Resolutions and Reduced Filter Orders (All errors in dB).

	16/14-Bit Set			
POLE PAIRS PER RECEIVE FILTER:	8	10	12	16
ERROR @ -7 dB STIMULUS LEVEL:	0,229	0,073	0,040	0,023
ERROR @ -13 dB STIMULUS LEVEL:	1,956	0,268	0,054	0,018
ERROR @ -43 dB STIMULUS LEVEL:	0,998	0,132	0,029	0,006

	14/12-Bit Set			
POLE PAIRS PER RECEIVE FILTER:	10	12	14	16
ERROR @ -7 dB	0,068	0,035	0,025	0,019
ERROR @ -13 dB	0,285	0,072	0,043	0,036
ERROR @ -43 dB	0,132	0,029	0,012	0,007

	12/12-Bit Set			
POLE PAIRS PER RECEIVE FILTER:	10	12	14	16
ERROR @ -7 dB	0,065	0,031	0,020	0,015
ERROR @ -13 dB	0,289	0,076	0,048	0,041
ERROR @ -43 dB	0,143	0,040	0,024	0,018

4.7 Summarising Comments on the Simulations Performed

The simulations described in this chapter have demonstrated the ease with which precise quantitative studies may be completed using the simulator. The simulations described illustrate ideas in three areas : the optimal SQNR performance attainable by a perfect implementation of a PCM system based on the (DAC-76) CCMDAC, the design of a suitable test stimulus sequence, and the design of a practical test set based on the dominantly digital approach proposed in Section 2.5. The simulations performed should be regarded as initial studies only, as too few situations have been investigated for definite general rules to be suggested.

In regard to test sequence design, it appears that it may be necessary to use stimulus lengths greater than one second. Nevertheless, the required length is likely to be practicable. It appears that the SQNR characteristic measured is dependent upon the stimulus length and frequency domain characteristics, with particularly great dependence within the region of high stimulus level (7dB down on peak input).
In regard to practical test set design, it has been found that the data converter resolutions required are consistent with those of available converter circuits. The stimulus generation section of the test set may be readily implemented. It has been found, however, that the receive side filtering effort required is substantial. Further simulation work is desirable in order to establish how much the cost of this filtering can be reduced.

As reported in Section 4.1, it has not been possible to generate narrowband stimuli with accurately speech-like PDF. All stimuli generated have approximately Gaussian distribution. The sufficiency conditions for the occurrance of this limiting effect, and the significance of the effect are discussed in Chapter 5.

5 FURTHER PROPOSALS AND CONCLUSIONS

5.1 The Response of Linear Filters to Stimuli which have Non-Gaussian Probability Distribution

The simulations of Section 4.1 suggest that the PDF of the output of the send filter tends to a Gaussian distribution and is largely independent of the PDF of the input signal. In order to conclusively determine the usefulness of prescribable distribution noise sources in QD measurement, it is necessary to determine whether the above effect can be avoided in QD measurement work. It is necessary to determine precisely what conditions are sufficient to ensure that the limiting effect will occur. It may be that these conditions have been introduced by the specific filter design procedure which happens to have been used, or that there do exist, in fact, input distributions which will give the required output distributions.

The purpose of this section is to discuss some aspects of the general analytic problem of determining the output of a linear network which is excited by a non-Gaussian source. It is felt unlikely that the problem is close to solution. Some common analytic approaches will be explored here. In Section 5.2, an algorithm is proposed to obtain precise SQNR values corresponding to any prescribed stimulus PDF.

5.1.1 General Analytic Techniques for Systems with Non-Gaussian Stimuli

Considering the present work in general terms, there is a need for analytic techniques to characterise the output of a system in terms of a stochastic input to the system. It has been arbitrarily decided in the present work to consider only the characterisation of the output probability distribution of a linear system. It is specifically noted that an exhaustive survey of the possibly relevant literature has not been done. No complete formal analysis of a system with arbitrary input PDF has been found. Davenport and Root (1958, p 189) have treated the problem with outstanding conciseness and honesty. The major section of their treatment is quoted below.

'If the input random process is non-Gaussian, the probability distributions characterizing the output process are not, in general, determined by the autocorrelation and mean of the output.

This being so, it is natural to ask what can be done toward obtaining output probability distributions for a linear system driven by a non-gaussian random process. The answer is, usually very little. There seems to be no general method for finding even a single-variate probability distribution of the output, beyond the brute-force method of calculating all its moments. This is unsatisfactory, of course, first because not all distributions are determined by their moments, and second because it is usually a practical impossibility to get all the moments in an easily calculable form. However, a few examples of this type of problem have been worked out by one device or another; in most of these examples the input random process, although not gaussian, is some particular non-linear functional or class of functionals of a gaussian process, such as the envelope, the absolute value, or the square.'

Before embarking fanatically on an exploration of the unpromising conventional procedures like moment computation, it is sensible to consider the applicability of broad principles such as the central limit theorems (Cramer, 1974, p 213 et seq).

5.1.2 The Applicability of the Central Limit Theorems

The idea here is to consider whether the filter output PDF is not constrained by a central limit theorem, and to find the most general sufficiency conditions for the theorem to apply to the random source/filter combination.

The most commonly cited central limit theorem is the Lindeberg-Levy theorem, or the equal components case. It is not difficult to show that this theorem cannot be applied to the random source/filter situation. The reasoning is outlined below. Considering also the possibility of recursive filters, the filter algorithm in general may be written as :

$$y_{k+1} = \sum_{i=0}^{k+1} a_i . x_{k+1-i} + \sum_{i=0}^{k} b_i . y_{k-i} \qquad (5.1\text{-}1)$$

where the samples x_i form the filter input time sequence, and the samples y_i form the filter output sequence. The filter coefficients a_i and b_i will be considered here to be constants.

When a detectable amount of send filtering is performed, the output signal Y will not have a uniformly flat power density spectrum. By application of the Wiener-Khintchine theorems, it is clear that the autocorrelation of Y will not be an impulse function. Hence the samples y_i will not be linearly independent. Davenport and Root (1958, p 58) have shown that linearly independent variables are not necessarily statistically

independent. However it is readily shown that linearly dependent variables cannot be statistically independent. Statistical independence is a pre-requisite for the application of most central limit theorems, and so the formulation of equation (5.1 - 1) is not convenient. Successive substitution of equation (5.1 - 1) allows the general recursive form to be decomposed to a non-recursive form.

$$y_{k+1} = \lim_{F \to \infty} \sum_{i=0}^{F} C_i \cdot x_{k+1-i} \qquad (5.1\text{-}2)$$

It is readily shown that this series is convergent if the filter is stable. It follows that any stable filter of the form of equation (5.1 - 1) can be approximated to any desired accuracy by a filter of the form:

$$y_{k+1} = \sum_{i=1}^{N} C_i \cdot x_{k+1-i} \qquad (5.1\text{-}3)$$

The approximation corresponds to truncating the impulse response, and the purpose is to give a series with a finite number of terms.

Let us now consider the form of the central limit theorems. Cramer's statement of the theorems (1974, p 214) is given below.

'Whatever be the distributions of the independent variables $[x_i]$ - subject to certain very general conditions - the sum $[x_s = x_1 + \ldots\ldots + x_n]$ is asymptotically normal ...'

Consider now the equal components case (Lindeberg - Levy theorem). The proof of the theorem relies upon the variables x_1 to x_n all having the same characteristic function, i.e. all having the same distribution. Now as the coefficients C_i are not all equal, in filters of interest, it is clear that each term in the sum is drawn from a different distribution. Hence the Lindeberg - Levy theorem cannot be applied to the situation under consideration.

By introducing restrictions additional to those limiting the validity of the Lindeberg - Levy theorem, it is possible to find central limit theorems applying to finite sums where each term of the sum is drawn from an independent distribution. Of such theorems, the Liapounoff theorem is

possibly least restricted in its applicability. The explicit requirements for this theorem are drawn from Cramer's treatment (1974, p 213-216). Consider a random variable X. Let the normal distribution function be $\phi(x)$, defined by

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-t^2/2}\, dt \qquad (5.1\text{-}4)$$

If two quantities M_o and σ_o can be found such that the variable $\left(\frac{X - Mo}{\sigma_o}\right)$ tends to $\phi(x)$ as $n \to \infty$,

then X is said to be asymptotically normal (M_o, σ_o).

It is interesting to consider the precise requirements for the applicability of the Liapounoff theorem. Let $V_1, V_2, \ldots\ldots, V_n$ be independent random variables, and let M_i and σ_i be the mean and standard deviation of V_i.

A first requirement is that the third absolute moment of each V_i about its mean should exist in the sense that it is finite. Thus for a V_i,

$$\rho_i^3 = E\left(|V_i - M_i|^3\right) \qquad (5.1\text{-}5)$$

where $E(\bullet)$ denotes the expectation.
If also, the condition is satisfied that

$$\lim_{n \to \infty} \left\{ \frac{\left[\sum_{i=1}^{n} \rho_i^3\right]^{1/3}}{\left[\sum_{i=1}^{n} \sigma_i^2\right]^{1/2}} \right\} = 0 \qquad (5.1\text{-}6)$$

then $$V = \sum_{i=1}^{n} V_i \qquad (5.1\text{-}7)$$

is asymptotically normal.

The central limit theorems refer to the (cumulative) distribution function of the sum. The conditions are known under which asymptotic normal convergence of the distribution function implies that the frequency function (PDF) tends to the normal PDF, but fulfilment of these conditions is assumed in the present argument. The conditions of equations (5.1 - 5) and (5.1 -6) will be referred to subsequently as the Liapounoff condition.

5.1.3 The Applicability of the Liapounoff Theorem

It was shown in the previous section that the equal components case of the central limit theorems is not applicable to the random source/filter situation. The conditions for applicability of the Liapounoff theorem have been stated. The purpose of this section is to examine the applicability of the Liapounoff theorem to the source/filter combination.

It is natural to ask whether fulfilment of the Liapounoff condition is not ensured simply by the constraint that the filter be stable. However, stability is not a sufficient condition. Some particular examples of stable filters have been constructed where the output PDF is demonstrably not asymptotically normal. Consider a filter fed from a random source with N - bit output wordlength, for example.

Let M =

Consider a 'filter' defined by

$$y_{k+1} = \sum_{i=0}^{n} \left(\frac{1}{M}\right)^{i} \cdot X_{k+1-i} \qquad (5.1\text{-}8)$$

It may be rigorously demonstrated that if the random source output is uniformly distributed, the filter output PDF is also uniform. This holds true even in the limit as $n \rightarrow \infty$. It is emphasised that equation (5.1 - 8) is an example of a stable linear filter to which the Liapounoff theorem does not apply.

It is, however, readily possible to synthesise a counter example of a stable linear filter to which the Liapounoff theorem does apply. An obvious example is the averaging filter given by equation (5.1 - 9).

$$Y_{k+1} = \sum_{i=0}^{n} \frac{1}{n+1} \cdot X_{k+1-i} \qquad (5.1\text{-}9)$$

Hence it is clear that some stable linear filters will necessarily have Gaussian output PDF, while others will not. The characterisation of the filter output PDF by the above effect is, at the least, determined by the rigorous applicability of the Liapounoff theorem. The theoretical basis for QD measurement by non-Gaussian stimuli is a precise formulation of the conditions under which the Liapounoff condition is met. The aim of the remainder of this section is to provide this formulation in terms relevant to QD measurement.

Firstly, it is verified that the applicability of the Liapounoff theorem is determined solely by the filter coefficients C_i in equation (5.1 - 3). The Liapounoff condition is formulated below in terms of the filter coefficients. The purpose is to seek an understanding only, not rigorous conviction.

Let us compare the form of the filter output (5.1 - 3) with that of the sum in Liapounoff's theorem (5.1 - 7). We consider

$$V_i = C_i \cdot X_{k+1-i} \qquad (5.1\text{-}10)$$

The X subscript denoting position in the input time series may be replaced by a subscript indicating its role in forming the sum. Conveniently written,

$$V_i = C_i \cdot X_i \qquad (5.1\text{-}11)$$

Any direct current (DC) component of the stimulus may be separated out because the filters considered are linear.

$$m_i = E(V_i) = E(C_i.X_i) = 0 \qquad (5.1\text{-}12)$$

Hence the third absolute moment is, from (5.1 - 5),

$$\rho_i^3 = E(|C_i.X_i|^3) = |C_i|^3.E(|X_i|^3) \qquad (5.1\text{-}13)$$

And the variance is

$$\sigma_i^2 = E(|C_i.X_i|^2) = |C_i|^2.\ E(|X_i|^2) \qquad (5.1\text{-}14)$$

Now while the sum component variables V_i are drawn from different distributions, the inputs X_i are all drawn from the same distribution.

$$E(|X_i|^3) = E(|X_k|^3) \quad \text{and} \quad E(|X_i|^2) = E(|X_k|^2) \qquad (5.1\text{-}15)$$

for all combinations of i and k in the sum.

For any practicable random source, $|X|$ is finite and hence ρ exists, satisfying part of the Liapounoff condition (5.1 - 5). The remaining of the Liapounoff condition, (5.1 - 6), may now be rewritten in releva:t form. Let us denote

$$\lim_{n\to\infty} \left(\frac{\left[\sum_{i=1}^{n} \rho_i^3\right]^{1/3}}{\left[\sum_{i=1}^{n} \sigma_i^2\right]^{1/2}} \right) = L$$

Using the previous equations of this section, gives

$$L = \lim_{n\to\infty}\left[\frac{\left[|C_1|^3+|C_2|^3+\ldots+|C_n|^3\right]^{1/3}}{\left[|C_1|^2+|C_2|^2+\ldots+|C_n|^2\right]^{1/2}} \cdot \frac{E(|X|^3)^{1/3}}{E(|X|^2)^{1/2}}\right]$$
$$= \lim_{n\to\infty} F.D \quad \text{(say, respectively)} \qquad (5.1\text{-}16)$$

The $\lim_{n\to\infty} F=0$ is sufficient to ensure applicability of the Liapounoff theorem. The term D involves absolute moments and so cannot be zero for a source with finite RMS output. Hence it has been shown that the Liapounoff condition is equivalent here to a constraint on the filter coefficients.

It is now surprisingly easy to demonstrate that there exist no stable filters which fulfil the Liapounoff sufficiency condition. The condition requires

$$\lim_{n\to\infty} \frac{\left[|C_1|^3+|C_2|^3+\ldots+|C_n|^3\right]^{1/3}}{\left[|C_1|^2+|C_2|^2+\ldots+|C_n|^2\right]^{1/2}} = 0 \qquad (5.1\text{-}17)$$

The numerator is a non-decreasing function of n. It may readily be shown that the denominator series is convergent, and hence that (5.1 - 17) cannot be satisfied for any stable filter. Referring to (5.1 - 3), it is clear that the C_i values are equal to the output response of the filter when the filter is excited by a unit impulse input. Convergence of the impulse response series is conveniently shown by the integral test (Boas 1966, p 8). Kuo (1966, p 293) has shown that, for any stable linear network with impulse response h(t),

$$\int_{-\infty}^{+\infty} |h(t)|\, dt < \infty \qquad (5.1\text{-}18)$$

Since h(t) is absolutely integrable, the series $|h_i|$ and $|C_i|$ are convergent.

Kuo (1966, p 293) has also given the final value theorem for stable linear networks, namely

$$\lim_{t \to \infty} h(t) = 0 \qquad (5.1\text{-}19)$$

To test convergence of the series $|c_i|^2$ or $|h_i|^2$, we may discard a finite number of initial terms, so that by (5.1- 19) we get

$$|h_i|^2 \leq |h_i| \qquad \text{for all } i>m \text{ , } m \text{ finite} \qquad (5.1\text{-}20)$$

The comparison test (Boas, 1966, p 7) may then be applied to the series $|h_i|$ and $|h_i|^2$, demonstrating that the denominator series of (5.1 - 17) is convergent. This is not surprising because the series $|h_i|^2$ represents the energy in the impulse response. However, it has been shown, surprisingly perhaps, that the Liapounoff sufficiency condition cannot be met. Hence the Liapounoff theorem gives no results for the noise source/linear filter situation.

5.1.4 A Necessary Condition for the Central Limit Theorems

The preceding parts of Section 5.1 have shown that various central limit theorems cannot be applied to the noise source/linear filter situation. Some details of the reasons why the theorems cannot be applied have been given. The purpose has been to give an appreciation of the nature of the effect observed in Section 4.1.

Based on this appreciation, and the equations already presented, it is possible to show a less informative but more conclusive result. The conditions considered in previous sections have been sufficiency conditions only. All sufficiency conditions have been encompassed by a sufficiency condition due to Lindeberg, (Feller, 1957, p 239). Feller has shown that the Lindeberg condition is also a necessary condition for any central limit theorem to hold (for independent variables). It is now shown that no central limit theorem applies to the noise source/linear filter situation.

Consider V to be a sum of n independent random variables, as in (5.1 - 7). Let the variance of the sum be denoted by S_n^2.

$$S_n^2 = \text{Var } (V) \qquad (5.1 - 21)$$

Feller's word statement of the Lindeberg theorem is quoted. 'The Lindeberg theorem implies that every uniformly bounded sequence of mutually independent random variables obeys the central limit theorem, provided, of course, that $S_n \to \infty$ $\left[\text{as } n \to \infty\right]$'.

It has been shown in Section 5.1.3 that S_n does not become infinitely large when any stable linear filter in used. Consequently, no central limit theorem applies.

Stearns (1969) has investigated a situation where pseudo-random white noise has been passed through a particular family of digital filters. He has apparently also found the filter output PDF to be approximately Gaussian, and, glibly citing Feller's theorem, has attributed this to the central limit theorem. Hence while the effect reported in Section 4.1 has apparently been observed previously, there remains no satisfactory explanation known to the author. The important point here is that a precise analytic formulation of the conditions necessary for this effect to occur would form the theoretical basis for QD measurement by non-Gaussian stimuli. Future work to produce this analysis is called for in so far as it produces a conceptual completeness in QD measurement ideas. As far as practical QD measurement by non-Gaussian stimuli is concerned, however, the algorithm introduced in the next section appears to be a good solution.

5 2 A Proposed Test Algorithm

5.2.1 Theoretical Basis

A Gaussian signal is the most conveniently generated narrowband pseudo-random test stimulus. Notwithstanding the inconclusive analytic results of Section 5.1 for stationary statistical processes, it is possible to generate narrowband pseudo-random stimuli with any desired PDF. The purposes of this section are to show how an arbitrary PDF may be synthesised approximately and to propose an algorithm by which this synthesis need not be done explicitly.

Let us consider a situation where it is convenient to generate N signals. Let these signals be $X_i(t)$, and let $f_i(x)$ be the PDF corresponding to $X_i(t)$. Consider now that we form a signal $X_s(t)$, such that

$$X_s(t) = \begin{cases} X_1(t) & \text{for } 0 \le t \le T_1 \\ X_2(t) & \text{for } T_1 \le t \le (T_1 + T_2) \\ X_3(t) & \text{for } (T_1 + T_2) \le t \le (T_1 + T_2 + T_3) \\ \vdots & \\ X_n(t) & \text{for } \sum_{j=1}^{N-1} T_j \le t \le \sum_{j=1}^{N} T_j \end{cases}$$

$$\text{and } X_s\left(t + \sum_{j=1}^{N} T_j\right) = X_s(t). \qquad (5.2-1)$$

Let the PDF corresponding to $X_s(t)$ be $f_s(x)$.

Then

$$f_s(x) = \frac{\sum_{i=1}^{N} T_i . f_i(x)}{\sum_{i=1}^{N} T_i} \qquad (5.2-2)$$

Also, letting $F_i(\omega)$ be the Fourier transform of $X_i(t)$, it appears that in cases where

$$|F_i(\omega)| \approx 0 \text{ for } k<|\omega|<h, \quad (k,h \in R \ge 0)$$

then

$$|F_s(\omega)| = 0 \quad \text{for} \quad k<|\omega|<h \qquad (5.2-3)$$

The approximate equalities become more exact with diminishing energy in signal transitions at $t = T_i$, i.e. as $\sum_{j=1}^{N} T_j \to \infty$ for finite N.

Given a PDF target $f_T(x)$, it is possible to synthesise $f_s(x)$ exactly, provided that the quantities on the right hand side (RHS) of equation (5.2 - 2) are not constrained. In cases of more practical interest, however, the designer requires only that $f_s(x) \approx f_T(x)$, and wishes to explore schemes whereby quantities in the RHS of equation (5.2 - 2) are conveniently constrained.

The mathematical environment of the above procedure is well known. The procedure has been stated in fairly general form in order to illustrate the myriad possible stimulus synthesis avenues. Let us now concentrate attention on one convenient avenue.

In the previous work it was found to be desirable that

$$f_T(x) = \frac{k}{\Gamma(h)} . (kx)^{h-1} . e^{-kx} \qquad (5.2-4)$$

where $x = v/v_{RMS} \ge 0$ and $k = \left[h(h+1)\right]^{1/2}$

$$\Gamma(h) = \int_0^{+\infty} x^{h-1} . e^{-x}\, dx \quad \text{on } h \geq 0 \text{ , and } f_T(-x) = f_T(x)$$

$h = 0,5$ (say),

and that it is convenient to generate

$$f_i(x) = \frac{1}{\sqrt{2\pi}\, s_i} . e^{(-x^2/2s_i^2)} \qquad (5.2\text{-}5)$$

for s_i a real parameter.

It will be found convenient, initially, to consider

$$T_i = T \text{ for all } i = 1, 2, \ldots, N. \qquad (5.2 - 6)$$

The band limiting required will be implicit.

One approach would now be to construct a stimulus $X_s(t)$ of period NT, and giving $f_s(x)$ as an N term approximation to $f_T(x)$.

However, it is proposed that there is a more efficient algorithm for $N \geq 1$. The procedure is as follows :

(a) The SQNR versus stimulus level characteristic for the PCM system is established using a Gaussian (say) stimulus of period T.

(b) This characteristic (smoothed curve) is used to compute the characteristic which would have been measured had one used a stimulus with PDF an N term approximation to $f_T(x)$. The computational procedure is defined by its equivalent synthesis procedure as given by equation (5.2 - 2). Taking equations (5.2 - 2) (5.2 - 4) (5.2 - 5) and (5.2 - 6), and using (say) a least mean square matching criterion, gives the mean square mismatch between $f_T(x)$ and $f_s(x)$:

$$E^2(N,S_1,S_2,\ldots,S_N) = \int_{-\infty}^{+\infty} \left[f_T(x) - f_s(x)\right]^2 dx \qquad (5.2\text{-}7)$$

For any fixed N, we wish to find a vector $R_N = (S_1, S_2, \ldots, S_N)$ such that

$$\left.\frac{d(E^2)}{dR}\right|_{R=R_N} = 0 \quad \text{and at which} \quad \left.\frac{d^2(E^2)}{dR^2}\right|_{R=R_N} \geq 0 \tag{5.2-8}$$

In the event that R_N is not uniquely determinable on the above basis, it is desirable to use this freedom to ensure that U is minimised; where $U = S_H - S_L$, we require R_N such that

$$\left.\frac{dU}{dR}\right|_{R=R_N} = 0 \qquad \text{and} \qquad \left.\frac{d^2U}{dR^2}\right|_{R=R_N} \geq 0 \tag{5.2-9}$$

for $S_H \geq S_j$ on $j = 1,2 \ldots, N$ and $S_H = S_k$ for some $1 \leq k \leq N$

and $S_L \leq S_j$ on $j = 1,2, \ldots, N$ and $S_L = S_k$ for some $1 \leq k \leq N$.

It is noted that, in equation (5.2 - 2), the uniqueness of R_N is not determinable by test of orthogonality of the various $f_i(x)$; the approach has not been to fix the $f_i(x)$ and determine optimal coefficients T_i. Clearly the Gaussian $f_i(x)$ are not orthogonal.

The requirement of equation (5.2 - 9) will be seen to arise in terms of the computational procedure described later.

Recalling earlier comments on the stimulus crest factor, it is noted that we require $f_S(x)$ to be matched to $f_T(x)$ only over the range (say) $0 \leq x \leq 4$ (corresponding to a crest factor of 12 dB).

The analytic determination of R_N is, surprisingly perhaps, intractable even for $N = 2$. It has been convenient to find the points R_N for various N by computer program. The functions $f_T(x)$ and $f_S(x)$ have been represented by sampled versions of 100 points each. A utility program FUNFIT(LEE) has been used as an implementation of a Powell minimisation algorithm. Compliance with equation (5.2 -9) cannot be guaranteed for the results. It is emphasised that it is a non-trivial problem in itself to minimise a quantity which is a function of several parameter 'dimensions'. Consequently, there is no guarantee that the R_N computed represent global minima of the error function, as opposed to local minima. Some results are given in Table 3.

Table 3 Standard Deviations of Gaussian Components to Best Approximate the Gamma Distribution (for h = 0,5).

N	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
2	0,315	2,493							
3	0,230	0,897	5,885						
4	0,189	0,668	1,159	11,01					
5	0,167	0,505	0,862	1,160	28,73				
6	0,153	0,408	0,695	1,092	2,069	143,4			
7	0,144	0,342	0,578	1,044	1,045	3,241	203,5		
8	0,138	0,299	0,486	0,806	1,165	1,182	5,415	91,89	
9	0,134	0,255	0,522	0,529	1,043	1,261	1,352	9,974	46,49

Table 4 Standard Deviations of Gaussian Components to Approximate the Gamma Distribution, where the Ratio of Successive Standard Deviations is Constant.

N	S_1	S_2	S_2/S_1 (dB)
2	0,315	2,493	17,95
3	0,227	0,967	12,58
4	0,193	0,569	9,39
5	0,175	0,415	7,49

Note: Table 4 is referred to on page 112.

The computational algorithm is best illustrated by means of an example.

Consider the case where N = 2 is sufficient.

(a) The PCM system is tested using (say) a Gaussian stimulus. The SQNR is determined at a number of suitably spaced stimulus levels.

(b) It may then be desirable to construct, from the above points, a smooth curve representing the SQNR versus Gaussian stimulus level characteristic. Let this function be denoted by $Q_G(v)$, where

$$Q_G(v_1) \text{ dB} = 20 \log_{10}\left(\frac{v_1}{n_1}\right) \text{ dB} \qquad (5.2\text{-}10)$$

for v_1 = RMS stimulus voltage, and n_1 = RMS noise voltage. The smooth curve should be constructed with the Q and v values expressed in decibel form.

(c) Noting that for N = 2, from Table 3, gives

$$\frac{S_2}{S_1} = \frac{2,493}{0,315} = 17,95 \text{ dB} \qquad (5.2\text{-}11)$$

Hence consider two (N = 2) points on $Q_G(v)$ such that

$$Q_G(v_1) = 20 \log_{10}\left(\frac{v_1}{n_1}\right) \qquad (5.2\text{-}12)$$

and

$$Q_G(v_2) = 20 \log_{10}\left(\frac{v_2}{n_2}\right) \qquad (5.2\text{-}13)$$

giving, for $v_1 > v_2$,

$$20 \log_{10}\left(\frac{v_1}{v_2}\right) = 17,95 \text{ dB} \qquad (5.2\text{-}14)$$

(d) Now, if we had a measurement stimulus v_s formed by concatenating v_1 and v_2 in time as defined by equation (5.2 - 1) we would get

$$Q_s(v_s)\ \text{dB} = 20 \log_{10}\left(\frac{v_s}{n_s}\right) \tag{5.2-15}$$

where $Q_s(v_s)$ = SQNR measured with approximately speech-like PDF. As all points $Q_G(v)$ were measured over the Gaussian source period T, this gives

$$v_s^2 = \tfrac{1}{2}\left(v_1^2 + v_2^2\right) \tag{5.2-16}$$

and

$$n_s^2 = \tfrac{1}{2}\left(n_1^2 + n_2^2\right) \tag{5.2-17}$$

(e) Using equations (5.2 -10) through (5.2 -17), it may be shown that

$$Q_s(v_s)\Big|_{v_s=v_e} = 10 \log_{10}\left\{\frac{1,01605}{10^{-(Q_G(v_1)/10)} + 10^{-(1,795 + Q_G(v_2)/10)}}\right\} \tag{5.2-18}$$

$$\text{for} \quad v_e(\text{dB}) = v_1(\text{dB}) - 2,94\ (\text{dB}) \tag{5.2-19}$$

Hence equations (5.2 - 18) and (5.2 - 19) may be used to compute the $Q_s(v_s)$ curve which would have been measured with a non-Gaussian and statistically non-stationary stimulus.

Equations (5.2 - 18) and (5.2 - 19) may be extended to give an effective stimulus PDF which is closer to the gamma distribution, by use of Table 3. This approach determines the error which would arise in SQNR measurement due to the use of a Gaussian stimulus as opposed to a more speech-like one. However, the computational algorithm suggests that there is little reason for incurring this error.

Equations (5.2 - 14) and (5.2 - 19) indicate that $Q_s(v_s)$ will be determined over a narrower domain than the $Q_G(v_G)$ measurements. $Q_G(v)$ should be measured at suitable points taking cognizance of the computational procedure to be followed subsequently. If N = 2 is sufficient, a first set of measurements would be made at 17,95dB stimulus intervals. A second set of measurements would be made with stimulus levels also 17,95dB apart, but with the initial point offset from the initial point of the first set, and so on.

For values of N > 2, and the algorithms suggested in Table 3, it is more difficult to find suitable sets of measurement stimulus levels. Hence it is attractive to obtain some values of $Q_G(v_i)$, for use in extensions of equation (5.2 - 18), by constructing a smooth curve through the sparse measured points on $Q_G(v_G)$. This smooth curve is best constructed to apply over a limited domain only.

It is felt that the reliable automatic operation of curve fitting routines cannot be guaranteed. Hence the possibility has been explored that the curve fitting exercise be obviated by constraining the PDF approximation algorithm to operate on regularly spaced stimulus levels. It gives PDF approximation algorithms which can be characterised by only two parameters (N and S_1/S_2).

Hence, again assuming equation (5.2 - 6) initially at least,

$$f_s(x) = \frac{1}{N\sqrt{2\pi}}\left(\frac{1}{S_1}\cdot \exp(-x^2/2S_1^2) + \sum_{i=2}^{N} \frac{1}{S_2\ S_2/S_1^{\,i-2}}\cdot \exp\left[-x^2/\left(2S_2^{\,2}(S_2/S_1)^{2i-4}\right)\right]\right) \qquad (5.2\text{-}20)$$

The notation assumes a monotonic variation of S_1 with i from 1 to N.

Clearly, the intention is again to avoid explicit synthesis of $f_S(x)$, and to use equation (5.2 - 20) only to define the computational procedure to transform measurable points on $Q_G(v)$ to points on $Q_S(v)$.

The determination of S_1/S_2 for various N has again been done numerically. The purpose being illustrative, only 100 points were again used for $f_S(x)$ and $f_T(x)$. The results are given in Table 4, (page 108).

The approximations suggested in Table 4 have been plotted; convergence to the gamma distribution with increasing N is, of course, very much slower than that obtained with the approximation parameters of Table 3. Consequently, initial attention has been concentrated on transforming sets of non-equispaced points. This is detailed in Section 5.2.2.

It is tempting to further refine the proposed algorithms for equally spaced stimulus points. It seems promising to determine a practically suitable value of N, by considering the number of measurements required. Having fixed S_1/S_2 by this N, more accurate computational procedures could be formulated to utilise all the $Q_S(v)$ points with suitable weighting of each point. However, as evidenced by Chapter 4, such refinement is not significant at this stage.

The proposed algorithm is of great importance in the present work because it demonstrates how the convenient Gaussian stimulus can be used to extract the required information from the PCM channel. In terms of this algorithm, and the results of Section 4.1, it is felt that the present work has been conclusive in fulfilling one of its aims. It has been established that sophisticated techniques for generation of arbitrary noise PDF's have no apparent application in the measurement of quantisation distortion.

5.2.2 Numerical Significance

A central question in the present work has been the following: how great is the error in measured SQNR due to the use of a Gaussian as opposed to speech-like test stimulus PDF? The purpose of this section is to quantitatively answer this question, at least in the case of the PCM system modelled.

Section 5.2.1 provides a basis on which to calculate the SQNR for a stimulus which is an N component approximation to the gamma distributed PDF. The difference between this calculated SQNR and the SQNR measured by Gaussian stimulus may be regarded as an SQNR error due to less suitable stimulus PDF, and this error will be denoted by $E_{PDF}(N,v)$.

With the purpose of calculating $E_{PDF}(N,v)$, the various transformation procedures for N non-equispaced components have been mechanised in a program XFMG2S, (Appendix H). Linear interpolation of measured points has been used where necessary. For each set of points used to calculate a transformed point, the numerically most significant point has been taken at a measured point. Hence interpolated values have been used only for the less numerically significant points. (Numerical values in this chapter are illustrative only). Again, the COMDAC model has been taken as representative, and the results of Section 4.2 have been used. These results have been transformed by XFMG2S for various N, and values of $E_{PDF}(N, v)$ are shown in Figure 31.

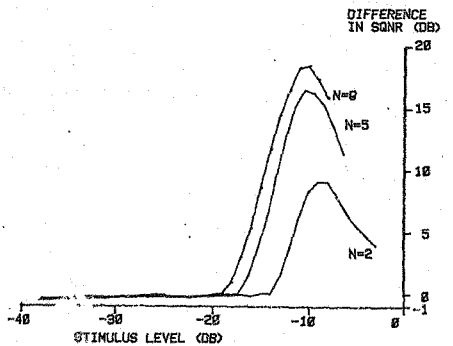


Figure 31 Differences in SQNR measured with Stimulus PDF's tending to a Gamma Distribution, Relative to Gaussian Stimulus Measurements.

Clearly the Gaussian SQNR curve is seriously misleading for stimulus levels in the range -10dB to -20dB. This is the range of greatest interest in determining how to load the encoder, and hence the results of Figure 31 are seen as very *significant*.

It is noted that the N-Gaussian approximation converges to the gamma distribution only at $N \geq 9$. The more meaningful criterion for convergence here is convergence of the *measured* SQNR with increasing N, rather than (say) a mean square error mismatch between the distributions. Increasing N from 8 to 9 does not affect measured SQNR by much more than 1dB anywhere within the stimulus level range ~10dB to ~20dB.

The SQNR curve for N=9 diverges from the Gaussian measured curve also at low stimulus levels. Errors of order 0,3dB occur for stimuli below -40dB, but this is of less interest. It is felt that useful results have been found to determine the significance of testing with Gaussian versus speech-like stimulus PDF, and that the present work is satisfactorily complete in this fundamental aspect.

5.3 A Proposal for a Deterministic Test Stimulus

5.3.1 Approach and Motivation

In the light of the conclusions of Section 5.2, it is appropriate to reconsider the test stimulus requirements stated explicitly in Section 2.3. It seems that the use of noise in previous work has been dictated merely by convenience. [illegible] are technology is now such that it may be convenient to take [illegible]ent approach. The purpose of this section is to outline an approach which may now be more hardware compatible than the *filtered noise approach*.

In the approach previously adopted, it is necessary to expend some effort to implement the send filter. For a hardware test set, it is unattractive to do this filtering in real-time. Two alternative implementations are possible. The entire test sequence may be held in read-only-memory (ROM). Alternatively, a directly analogous approach to that of the *Nova* package may be used. Here a block of random-access memory (RAM) is required, and this must also be large enough to accommodate the entire test sequence. The test sequence may be computed slowly and stored in the RAM, then replaced word by word by the receive side signal. This facilitates receive side filter implementation.

However, suffice it to note that the amount of filtering effort on the send side is about equal to that of both receive side filters together (Section 4.6.2). This is the motivation for proposing here an approach which obviates the need for send side filtering. Even if this approach is

found to be surpassed in hardware implementation by pre-stored stimuli in ROM, it is desirable to evaluate this method of stimulus sequence construction. It will be seen that band limited but non-Gaussian stimuli may be constructed directly by the proposed approach. The approach is to generate the stimulus so that it is inherently bandlimited. One suitable synthesis procedure might be to generate a stimulus x(t) such that:

$$x(t) = \sum_{i=0}^{N} A_i . \sin\left[(\omega_L + i.\Delta\omega)t\right] \qquad (5.3\text{-}1)$$

where $f_L = \omega_L/2\pi$ = lower limit of stimulus band

$f_H = \omega_H/2\pi$ = upper limit of stimulus band

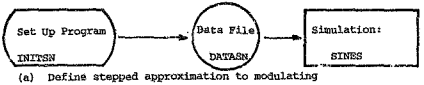
$\omega_H = \omega_L + N.\Delta\omega$

and N, ω_L and ω_H are design parameters.

Let $X(\omega)$ denote the Fourier transform of x(t), then it is clear that $|X(\omega)| = 0$ except at points on $\omega_L \leq |\omega| \leq \omega_H$ and hence no explicit send filter is required if $A_i \neq A_j(t)$. Let f(x) denote the function describing the distribution of amplitudes of x(t), analogous to the PDF if x(t) were a random signal. It is not attractive to seek to determine f(x) analytically in terms of the various A_i.

Each term $A_i.\sin(\omega_i t)$ represents a balanced amplitude modulation process, with well known sideband generation effects when $A_i = A_j(t)$. It is proposed that a suitable test stimulus could be synthesised with careful selection of design parameters N, ω_L, ω_H and $A_i(t)$. There are design constraints in order to make this a preferable approach to implement in hardware, as opposed to a filtered noise source. Hence N should be kept small, and a simple method found to generate the $A_i(t)$. For simplicity it is preferable that $A_i(t) = A_k(t)$ for all i and k; then the modulating signal may be denoted as A(t). It is required that A(t) be bandlimited to rather less than the stimulus bandwidth. This A(t), together with the selected N, ω_L and ω_H, should give a suitable f(x).

As the primary tool is such a design process, simulation is again attractive. A simulation package (SINES) has been written for this purpose and is listed in Appendix I. Only part of the structure of the package is outlined in Figure 32, as it incorporates some of the features of the filtered noise simulation package for the IBM 370.



(a) Define stepped approximation to modulating waveform.
(b) Bandpass filter if required.
(c) Generate sum of sinusoid sub-carriers.
(d) Modulate.
(e) Bandpass filter stimulus if required.

Figure 32 Outline of Stimulus Source Section of Simulation Package SINES for IBM 370.

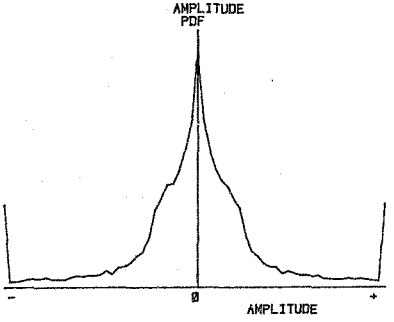
5.3.2 Results of Simulations to Gauge Feasibility

Initial simulations have been performed to gauge whether usable amplitude distributions can be achieved with simple modulating envelope waveforms. The aim has been either to approximate the gamma distribution directly, or to get a stimulus distribution which gives faster convergence in the implicit synthesis transformation procedure described in Section 5.2 (in place of the Gaussian components). It is convenient to loosely refer to the amplitude distribution density function $f(x)$ as a PDF.

Consider a 500 Hz sinusoid being sampled at 8kHz, with one sample point at a sinusoid zero crossing point. This sampling renders a picture of the sinusoid PDF which is not suggestive of the analytically derived sinusoid PDF. Two points emerge. Firstly, analytic calculation of SQNR should take account of any periodicities in the signal, as in the above case the encode characteristic is 'sampled' for quantisation step size at only four places (ignoring polarity). The more important point is that the sampled PDF is the one that the PCM system actually 'sees'.

An initial set of simulations was done with all $A_i(t)$ equal to a constant. Spiky PDF's were found when the sinusoidal sub-carriers were symmetrically distributed about the 500Hz sub-harmonic of the sampling frequency. A better set-up is to have sub-carrier extremes at 410Hz and 490Hz, so that the test stimulus lies within 400Hz and 500Hz even allowing for a non-constant modulating envelope. It is also helpful to avoid even numbers of sub-carriers, and $N = 31$ is conveniently prime and within practical limits of real-time generation. This gives a PDF shown in Fig. 33. Not too surprisingly, this PDF is closer in form to the speech PDF than is the Gaussian.

A further example of a PDF is shown in Figure 34. This was achieved by using a modulation envelope defined to have the value 1000 for 1 second and then 4500 for the next second. This envelope was bandlimited to 8Hz



NOTE: EXTREME AMPLITUDE PEAKS IN PDF CAUSED BY SIMULATED D/A CONVERTER CLIPPING.

FIGURE 93 AMPLITUDE DISTRIBUTION OF NARROWBAND STIMULUS SYNTHESISED FROM SINUSOIDS.

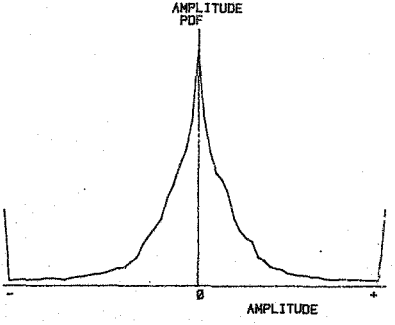


FIGURE 94 SPEECH-LIKE DISTRIBUTION OF STIMULUS SYNTHESISED FROM MODULATED SINUSOIDS.

using an eight pole Butterworth filter prior to modulation. It is proposed that further work would be useful to find a readily implemented scheme to generate the bandlimited envelope.

A more complex envelope is readily found to give an approximately Gaussian PDF. However, the non-Gaussian PDF's attainable by deterministic stimuli may be more suitable for building-up the gamma distribution.

5.4 Preliminary Comments on Hardware Test Sets

Before serious hardware design can be undertaken, it is desirable that a few remaining points be investigated. These points include mainly the use of filter types more suitable than the Butterworth, suitable modulation generation for the deterministic stimuli proposed in Section 5.3, and determination of the frequency spectral and statistical properties required of the stimulus to give good correlation between measured and subjective evalutions, as discussed in Section 2.3.

Hence the hardware sources outlined in Figures 35 to 37 are speculative.

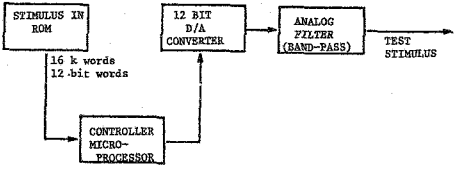


Figure 35 Source with Stimulus Stored in ROM.

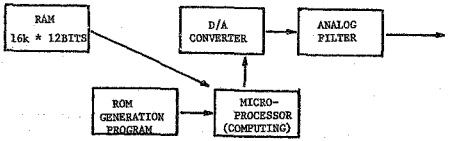


Figure 36 Source with Off-line Calculation of Stimulus Sequence.

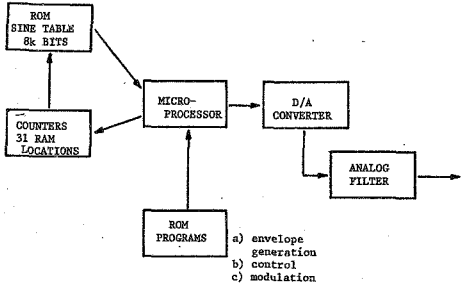


Figure 37 Deterministic Stimulus Source.

Erasable 3708 PROM's are currently available at about R10 for 8k-bits. Hence the EPROM cost in Figure 35 would be about R240, but only a modest single chip microprocessor would be required. Also, the ROM cost would be less if non-erasable types were used.

Referring to Figure 36, the feasibility of generating the stimulus sufficiently fast has not yet been proved, but the filtering is about equivalent to that necessary on the receive side for any system. Filtering may be done more slowly, and the RAM might be required in the receive section anyway. The limit on test duration would be around eight seconds.

The sinusoid look-up table of Figure 37 may be implemented in only 8k-bits of ROM, (Barna and Portat, 1973, p 408). This approach would be the most attractive if the receive side filtering could be done in real-time. The two receive side filters could easily be implemented in parallel, and each structure is inherently susceptible to pipeline architecture. This is, as yet, merely a proposal for future work.

The purpose of this section has been to suggest some perspective in terms of hardware, and further speculation is unwise.

5.5 Noise Sources for Future Work

As indicated in Section 2.3.2, an early constraint in the present work was to investigate the usefulness of Walker's zero-order Markov source techniques in QD measurement. It is felt that this specific investigation is now sufficiently definite to be regarded as complete. Walker (1978) has recently described techniques for n-th order Markov sources with prescribed PDF. These techniques are particularly suited to the construction of hardware sources. Future work on QD measurement stimuli will necessarily have to consider not only Walker's recent techniques which allow control of both PDF and second order statistical properties, but also the conceptually distinct techniques of Broste (1971) and those of Polge, Holliday and Bhagavan (1973).

Walker's technique involves transition probability sampling as an extension of the technique adopted in the present work. Compared with the very stringent stimulus power spectral specifications evolved in the present work (Section 4.6), only very limited control of the noise power spectrum has thus far been demonstrated by Walker to be practicable (Walker, 1978). There appear to be no present indications that the real-time computation of stimulus sequences will become cost competitive with stimuli stored in ROM. Consequently, future work will need to admit for consideration techniques which are computationally far clumsier than Walker's, but which hold out prospects of rigid spectral control.

Broste (1971) and [illegible]i and Hammond (1975) have used the different technique of 'operating' on a vector of independent Gaussian variables first by a linear transformation matrix and then by a non-linear transform'. It is transparent that this is a process of linear filtering to shape the power spectrum, and then a non-linear gain to distort the Gaussian PDF to the desired PDF. It is felt that this technique is unlikely to be very useful for test stimulus generation because of the need here to limit signal bandwidth strictly. However, future work would usefully establish this point rigorously.

Polge et al (1973) have described a different technique called correlation transfer. The technique requires two sets of sample values to be generated. One set is an ordered set having the desired probability

distribution, but with the sample values arranged in non-decreasing order. This set is readily constructed. The second set is required to exhibit the correlation properties prescribed for the ultimate output sequence, but may have any convenient PDF. The correlation properties of the second set are then transferred to the first by re-ordering the values in the first set. This method warrants further consideration in stimulus generation studies. Much of the software necessary to quantitatively evaluate this transfer technique has been provided by the present work. The second set mentioned above corresponds directly to the stimuli generated in the present work. It appears, however, that the correlation transfer process described is 'loose' in that the correlation properties are merely induced in the first set, rather than directly forced upon it.

It is emphasised that the above investigations are motivated by a desire for conceptual completeness in defining the solution to the QD measurement stimulus problem. The practical solution will almost certainly be based on the Gaussian stimulus algorithm proposed in Section 5.2.

5.6 The Test System Sampling Frequency

In Section 2.5.2 it was shown that the analog filter following the stimulus generating D/A converter could sometimes be omitted from the test set. This stimulus reconstruction filter was conceptually inserted in the simulator packages by locking the test set sampling frequency to the sampling frequency of the PCM system. All the simulations of Chapter 4 were performed using a common sampling frequency - nominally 8kHz.

The common clock and implicit analog filter are very convenient for laboratory work. In future work where a dedicated hardware test set is closer to finalisation, however, it is necessary to incorporate analog filter circuitry as motivated in Section 2.5.2. Furthermore, it would then be wise to avoid any limitations of having access to, or knowledge of, the PCM sampling frequency. An anti-aliasing filter at the input of the receive side of the test set is anticipated, incidentally.

More importantly, however, the test set sampling frequencies are then available as design parameters. On the receive side, the sampling frequency is determined by the upper limit of the noise measurement band. This limit is dictated by the lowest PCM system sampling frequency with which the tester will be expected to cope. For speech link testing, the test set receive side sampling frequency cannot sensibly be made much less than 8kHz. This parameter cannot be used to reduce the computational effort of the receive side filtering.

In generating the narrow bandpass stimulus, however, it is possible to reduce filtering computation by reducing the sampling frequency to some 1,5kHz rather than 8kHz. This has not been done in the present work, and it is felt that there would be disadvantages of this approach in future work.

A reduction in stimulus generator sampling frequency is a trade-off of digital filtering effort for analog filtering effort. The analog filter constructing the test stimulus from the D/A converter output does not have demanding specifications if an 8kHz sampling frequency is retained. On the other hand, a high sampling frequency merely affects the amount of ROM needed, assuming storage of a precomputed stimulus. ROM is relatively cheap.

A high sampling frequency in the stimulus generator can also be used to reduce D/A converter requirements. Cattermole (1969, p 153-158) has extended Bennett's analysis (1948) of the combined effects of sampling and quantising, and gives results for encoders with uniform quantum size. As indicated in Section 3.1.2.5, there is no conceptual difference between a D/A converter operating on finely quantised data and an A/D converter operating on a continuous signal. Now, it is clear from Cattermole's analysis that the total quantisation noise power is determined roughly by the data converter resolution. This noise power may be spread more 'thinly' by increasing the sampling frequency. Sampling above the Nyquist rate enables the inband noise power to be reduced, and an analog filter can be used to improve the SQNR. There is a limit to this improvement as the oversampled signal tends towards a continuous time signal with only quantisation impairment; but more than 10dB improvement is possible. Within this limit, doubling the sampling frequency will improve the D/A converter SQNR by 3dB, as intuitively expected.

In the simulations described in Section 4.5, the stimulus band was 450Hz to 550Hz and the sampling frequency 8kHz. The test set back-to-back SNR had a peak value of 62,5dB; and this is close to the minimum requirement of 63dB. Now if the set simulated were to incorporate a non-critical analog bandpass filter, the high sampling rate could readily give a SQNR benefit equivalent to an extra bit of D/A resolution (6dB). It was on this basis that a 12-bit resolution was accepted for the hardware D/A converter.

There is the possibility of making the hardware test set frequency scalable under user control. The analog filters would have to be roughly tunable for various digital section sampling frequencies. Further work is necessary to evaluate the usefulness of such a feature.

5.7 Concluding Comments

The purpose of this brief section is to summarise in retrospect the present work. It is felt that the present work has made a start towards the following activities.

(a) Formally establishing the nature of the quantisation distortion measurement problem in PCM systems, by reviewing the relevant literature available and stating the measurement requirements explicitly.

(b) Proposing that time domain subtraction stimuli be narrowband also.

(c) Provision of simulation tools to investigate aspects of QD measurement systems. A useful Fortran/assembly language interface has been proposed.

(d) Establishing the small importance to QD measurement of sophisticated pseudo-random source techniques for prescribed distribution.

(e) Establishing design parameter values to roughly outline a feasible test system based largely on digital technology.

(f) A deterministic test stimulus has been proposed, and initial simulation tools provided to evaluate this.

(g) The generation of gamma distributed rather than accurately Gaussian stimuli has been proposed. The difficulties of generating such stimuli have been illustrated, and the error incurred by the use of a Gaussian stimulus has been calculated.

(h) A technique has been proposed whereby convenient (Gaussian) stimuli may be used for measurements; the performance for gamma distributed signals may then be inferred without explicit synthesis of these signals. This is the most important section of the present work.

(i) Embryo facilities have been established for laboratory hardware tests. Software and hardware versions of an encoder have been constructed to enable measurement of noise other than that due to quantisation.

(j) The non-applicability of the central limit theorems to filtering of non-Gaussian signals has been suggested and demonstrated. This has posed a fundamental question as to the theoretical basis of QD measurement.

It is expected that many of the concepts of QD measurements will find wider application in distortion measurement of completely analog systems. Cross-over distortion in amplifiers is not well measured by sinusoidal stimulus, for example.

I have been much encouraged by those various people who have shown a continued interest in this work, and I thank them.

BIBLIOGRAPHY CITED

Ball Computer Products Inc. 'DDOS 1.6 programmer's reference manual', 86-0140050-4, (1974).

Ball Computer Products Inc. 'DDOS programmer's reference manual addendum 1', 86-0140070-0, (1975).

BARNA, A. and PORTAT, D.I. 'Integrated circuits in digital electronics', Wiley, (1973).

BELCHER, R.A. 'A new distortion measurement - better subjective - objective correlation than given by THD', Wireless World, Vol. 83, 36 - 41, (May 1978).

BENDAT, J.S. and PIERSOL, A.G. 'Measurement and analysis of random data', Wiley, New York, (1966).

BENNETT, W.R. 'Spectra of quantised signals', Bell System Technical Journal, 27, 446 - 472, (July 1948).

BOAS, M.L. 'Mathematical methods in the physical sciences', Wiley, New York, (1966).

BROSTE, N.A. 'Digital generation of random sequences', IEEE Trans on Automatic Control, AC - 16, No. 2, 213 - 214, (April 1971).

CATTERMOLE, K.W. 'Principles of pulse code modulation', Iliffe, London, (1969).

COHEN, C.L. 'Audio world gets the word: digital', *Electronics*, 78-79, (Nov. 24, 1977).

COVEYOU, R.R. and MacPHERSON, R.D. 'Fourier analysis of uniform random number generators', J.ACM, 14, 110-119, (1967).

CRAMER, H. 'Mathematical methods of statistics', Princeton University Press, Princeton, (1974).

DATA GENERAL CORP. 'Extended basic user's manual', 093-000065 - 01, Massachusetts, (1972).

DATA GENERAL CORP. 'Fortran iv run time library user's manual', No. 093-000068, 5th Rev., (Feb. 1975).

DAVENPORT, W.B. AND ROOT, W.L. 'An introduction to the theory of random signals and noise', McGraw-Hill, New York, (1958).

DECISION INC. 'Decision disk operating system fortran', No. 87-0101912-1, U72-17.2, (Nov. 1972).

FELLER, W. 'An introduction to probability theory and its applications', Vol. 1, Second Edition, Wiley, New York, (1957).

GOLD, B. and RADER, C.M. 'Digital filter design techniques in the frequency domain', Proc. IEEE, 55, 149-171, (Feb. 1967).

GRAHAM, N. 'Microprocessor programming for computer hobbyists', Tab Books, Pa., U.S.A., (August 1977).

GREENBERGER, M. 'An a priori determination of serial correlation in computer generated random numbers', Math Comput., 15,383-389, (1961).

GUPTA, S.C. and HASDORFF, L. 'Fundamentals of automatic control', Wiley, N York, (1970).

HANRAHAN, H.E. 'An extended form of the bandwidth-time law for digital measurements on noise', Research Report, Wits University, (1978).

(b) 'Algorithms for the measurement of mean square value from data compressed with the μ-255 law', record of Symposium on Digital Signal Processing, S.A. Inst. of Electrical Engineers, (June 1978).

HARMAN, W.W. 'Principles of the statistical theory of communication', McGraw-Hill, New York, (1963).

HARTLEY, M.G. 'Digital simulation methods', Peter Peregrinus, (1975).

KAISER, J.F. 'Design methods for sampled data filters', reprinted in RABINER, L.R. and RADER, C.M. (Eds) 'Digital signal processing', 20-34, IEEE Press, (1972).

KIKKERT, C.J. 'Measurement of quantisation distortion for random inputs', Proc. IREE (Australia), Vol. 32, No. 12, 459-464, (Dec. 1971).

KUO, F.K. 'Network analysis and synthesis', Wiley, New York, (1966).

LEWIS, T.G. 'Distribution sampling for computer simulation', Lexington, Massachusetts, (1975).

LI, S.T. and HAMMOND, J.L. 'Generation of pseudo-random numbers with specified univariate distributions and correlation coefficients', IEEE Trans. on Systems, Man, and Cybernetics, 557-561, (Sept. 1975).

McDONALD, R.A. 'Signal-to-noise and idle channel performance of differential pulse code modulation systems - particular applications to voice signals', Bell System Technical Journal, 45, 1123-1151, (Sept. 1966).

POLGE, R.J., HOLLIDAY, E.M. and BHAGAVAN, B.K. 'Generation of a pseudo-random set with desired correlation and probability distribution', Simulation, 153-158, (May 1973).

PRECISION MONOLITHICS INC. Data Sheet on the DAC - 76 companding digital to analog converter, (July 1976).

RAMSDEN, A.N. 'P.C.M. testing techniques', Marconi Instrumentation, Vol. 12, No. 4, 73-79.

RICE, S.O. 'Mathematical analysis of random noise', Bell System Technical Journal, 24,50, (Jan. 1945).

RICHARDS, D.L. 'Telecommunication by speech: the transmission performance of telephone networks', Butterworth, London, (1973).

ROLLS, R.G. and WEBSTER, P.R.M. 'Pseudo-random noise sources as stimuli for quantization distortion measurements on p.c.m. systems', Marconi Instrumentation, Vol. 14, No. 4, 89-96, (1974).

SRIPAD, A.B. and SNYDER, D.L. 'A necessary and sufficient condition for quantization errors to be uniform and white', IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-25, No. 5, 442-448, (Oct. 1977).

STANLEY, W.D. 'Digital signal processing', Reston, (1975).

STEARNS, S.D. 'Digital simulation of frequency limited random functions', Simulation, 37-41, (Jan. 1969).

TAUE, H. and SCHILLING, D.L. 'Principles of communications systems', McGraw-Hill Kogakusha, Japan, (1971).

WADDINGTON, D.E. O'N. (a) 'Quantisation distortion', Marconi Instrumentation, Vol. 12, No. 1, 14-17, (March 1969).
(b) 'Overall evaluation of p.c.m. systems', Marconi Instrumentation, Vol. 13, No. 6, 135-140, (Dec. 1972).

WALKER, A.J. (a) 'New fast method of generating random numbers with arbitrary frequency distributions', Electron. Lett., 10, 127-128, (1974).

(b) 'An efficient method of generating discrete random variables with general distributions', ACM Trans. on Math. Software, Vol. 3, No. 3, 253-256, (Sept. 1977).

(c) 'A fast flexible digital hardware implemented markov data source', Ph.D. Thesis, University of the Witwatersrand, 1978.

CONTENTS OF APPENDICES PAGE

APPENDIX A A Laboratory PCM System

A.1 Introduction to the COMDAC PCM System 133

A.2 Hardware Architecture 133

A.3 Modes of Operation 133

A.3.1 The 'Continuous' Mode 135

A.3.2 Mode Control Words 135

A.4 Circuit Description

A.4.1 Mode Switching 136

A.4.2 External Data Operation 139

A.4.3 Operation of the Converter Section 139

A.4.4 The Serial Data Tri-State Gating 141

A.4.5 The Protection Zeners 142

A.4.6 Attenuator Operation 143

A.5 System Timing 143

A.6 Software Readable Flags 144

APPENDIX B A 12-Bit Digital to Analog Converter for the Nova Computer

B.1 Introduction 155

B.2 Circuit Description 155

APPENDIX C A Revision of the Dual 10-Bit Digital to Analog Converter Facility on the Nova Computer.

C.1 Introduction 157

C.2 Third Channel Circuit Description 157

APPENDIX D Simulator and Hardware Testing Software for the Nova Computer.

D.1 PROBSPEC 160

D.2 ALICUT 162

D.3 RNGE N 163

D.3.1 RNGSUB 1 165

D.3.2 RNGSUB 2 165

D.4 FLTS 166

D.5 FILTØ 167

D.6 BUTTER 187

CONTENTS OF APPENDICES PAGE

D.7 RESP ... 169
D.8 QUANTO ... 170
D.9 TESTP ... 173
D.10 PLOTP ... 174
D.11 IDLPCM ... 175
D.12 CONVERT ... 176
D.13 FLTRS ... 176
D.14 FLTRM ... 178

APPENDIX E Simulation and Utility Software for the IBM 370 Computer

E.1 INIT ... 180
E.2 PSPEC ... 182

APPENDIX F The Combined Transformations used for Digital Filter Design ... 184

APPENDIX G Numerical Data and Results from Simulations

G.1 Default Values for Parameters Not Explicitly Defined ... 197
G.2 Probability Distribution Modelling ... 198
G.3 The Ideal COMDAC SQNR versus Stimulus Level Characteristic 202
G.4 Measured SQNR Variation with Test Length ... 203
G.5 Measured SQNR as a Function of Stimulus Frequency Domain Parameters
G.5.1 Measured SQNR Variation with Stimulus Centre Frequency 204
G.5.2 Measured SQNR Variation with Stimulus Bandwidth 205
G.6 Data Converter Signal Level to Optimise Test Set SNR ... 206
G.7 Tests with Constrained Data Converter Resolution and Filter Order
G.7.1 Constrained Data Converter Resolution ... 207
G.7.2 Constrained Data Converter Resolution and Filter Order. ... 208

CONTENTS OF APPENDICES

		PAGE
APPENDIX H	SQNR Curve Transformation Program: XPMG2S	209
APPENDIX I	A Simulation Program for Deterministic Stimuli: SINES	
I.1	A Program to Set Up the Simulation: INITSN	212
I.2	Simulation Program for Deterministic Stimuli: SINES	214

APPENDIX A: A LABORATORY PCM SYSTEM

A.1 Introduction to the COMDAC PCM System

The system described here was developed because the need arose to make measurements of the quantising error performance of a representative pulse code modulation (PCM) communications system.

In order to maximise the flexibility and convenience of these experiments, the system has been configured as a Nova computer peripheral. In so doing, it has become possible to create a system of sufficient flexibility to be of interest as a general purpose experimental tool, comparable to the other data converters in the Wits Acoustic Instrumentation rack. Consequently, an additional aim of the system is to allow maximum diversity of use of the central components.

The system is built around two COMDAC companding digital to analog converters. These converters, as well as skeletal applications circuits and descriptions of circuit operation, are outlined in the Precision Monolithics Incorporated data sheet DAC-76. The subsequent system description assumes reader familiarity with the DAC-76 data sheet.

A.2 Hardware Architecture

Because of the incorporation of two COMDAC's, the system may be conceptually divided into two data conversion channels. These will be referred to here as 'channel A' and 'channel B'. The channel assignments are such that channel A, for example, is always associated with a particular COMDAC.

Each channel is capable of operation as an analog to digital data encoder, or as a decoder, or as one of a number of other functions. A system design goal was to make each channel independently configurable to any circuit function within the set of capabilities. The configuration state of the system has been referred to as the 'mode'.

A.3 Modes of Operation

The system mode is determined by Nova software and is written from the Nova to a mode latch in the system. Each channel may perform one of the following functions :

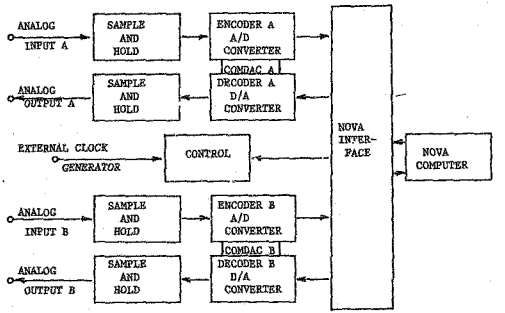


Figure A1 Functional Blocks in the PCM System Hardware.

(a) Companding encode of analog input to digital data, with 8 bit output wordlength. Digital data may be read into the Nova in parallel from a latch at the end of each sample conversion. Alternatively, it is available as serial data during the successive approximation conversion.

(b) Companding decode of digital data to analog sample and hold output, with 8 bit data wordlength. Digital data may be output in parallel from the Nova to a decoder latch. Alternatively, the data may be accepted serially, say from the other channel acting as an encoder.

(c) Extended range companding decode, with 9 bit wordlength. Digital data must be output in parallel form from, say, the Nova.

(d) Digitally controlled analog attenuator, with logarithmic step spacing. Attenuator setting is latched from a parallel data source of up to 9 bit wordlength.

A.3.1 The 'Continuous' Mode

For each of the encoding or decoding functions, the conversion may be triggered either by external clock generators, or by Nova input/output control signals under program control. The particular triggering source need not be specified in the mode control word, except when neither of the above classes of trigger signal is to be used. This occurs when it is desired that a channel should act as an encoder which starts to convert the next sample as soon as it has finished encoding the present sample. This 'continuous' mode allows the system to run at its maximum conversion rate, which is just over 50 000 samples per second.

The continuous mode is very convenient for initial debugging of the system. Also, the continuous mode may be used to generate an 'autonomous' PCM system for cases where only the analog input and output need be accessible.

It should be noted, however, that the continuous mode triggers both input sample and hold channels at the same time, and both output sample and hold channels at other times. Hence if one channel is set to continuous operation, then there are sensible restrictions on the operation of the other channel.

A.3.2 Mode Control Words

The first step in using the system is to load the 8 bit mode control latch with the desired mode control word. A four bit per channel mode representation was chosen partly because it facilitates user specification of desired mode, but primarily because it yields economical but logically segmented hardware.

The mode control word is made up as follows :



EXTERNAL DATA (DECODE) B
SEND (ENCODE) B
CONTINUOUS OPERATION B
ATTENUATOR B
EXTERNAL DATA A
SEND A
CONTINUOUS A
ATTENUATOR A

The options required are selected by making those bits logical "ones". Some control bits can override the effects of other control bits, as will be apparent later. A table of some modes is provided and this should suffice for most applications.

A.4 Circuit Description

Circuit diagrams of the system are given in Figures A2 through A12. These diagrams serve only to describe the system. Construction and maintenance details are detailed in the Wits Acoustics Instrumentation Rack laboratory handbook. Where a section of the circuitry is shown for one channel only, it may be assumed that the other channel is similar. The headings 'Analog Board', 'Digital Board One', and 'Digital Board Two' refer to the physical boards on which the circuitry has been built.

A.4.1 Mode Switching

The mode latch and associated circuitry is shown in Figure A5. An inverted representation of the mode control word is placed on the Nova data bus lines $\overline{\text{DATA 8}}$ through $\overline{\text{DATA 15}}$. These lines are inverted again in the system, as shown, and applied as inputs to the mode control latch. These data are then loaded into the mode control latch on the negative edge of the DATO.C pulse from the Nova. Gate G5 ensures that only DATO.C pulses addressed to the experimental section device code by a low $\overline{\text{DEV.SEL}}$ are used.

Gates G1, G2, G3, and G4 were found necessary to 'deglitch' the DATO.C line and stop spurious mode changes. Gate G4 output will remain logically low unless a pulse appears on DATO.C which is longer than the delay through gates G1 and G2 in series.

It will be noted that only the information 'continuous A OR continuous B' is used, as is accessible at gate G9 output. This is a compromise of desired system flexibility, but was unavoidable because the board edge connector gives a limitation of 43 connections. This limitation was a

TABLE A1 Mode Control of the PCM Hardware

CONTROL WORD	OCTAL REPRE-SENT-ATION	CHANNEL A FUNCTION	CHANNEL B FUNCTION	CONVERSION TRIGGER	NOTES
00100010	42	Encode Analog Input	Encode Analog Input	Ext 1, Start	Both channel Trig. Simultaneously
01100010	142				
00100110	46	"	"	Continuous	"
01100110	146				
00100000	40	"	Decode Serial Data From 'A'	Ext 1, Start	PCM Link Serial Data, A to B
01100000	140	"	"	Continuous	"
00100100	44				
01100100	144				
00000010	2	Decode Serial Data from "B"	Encode	Ext 1, Start	PCM Link, Serial Data, B to A
01000010	102	"	"	Continous	"
00000110	6				
01000110	106				
00000000	0	Decode Serial Data Input	Decode Serial Data Input	Ext 1, Start	No Applications Envisaged
01000000	100	"	"	Continuous	"
00000100	4				
01000100	104				
00100001	41	Encode Analog Input	Decode Parallel Word From Nova	Ext 1, Start	Separate functions, or parallel Data Transfer PCM link A to B

TABLE A1 Mode Control of the PCM Hardware

CONTROL WORD	OCTAL REPRESENTATION	CHANNEL A FUNCTION	CHANNEL B FUNCTION	CONVERSION TRIGGER	NOTES
01100001	141	Encode Analog Input	Decode Parallel Word from Nova	Continuous	Separate Functions, or Parallel Data Transfer PCM Link A to B
00100101	45				"
01100101	145				"
00010010	22	Decode Parallel Word from Nova	Encode	Ext 1, Start	Separate Functions, or Parallel Data Transfer PCM Link B to A
01010010	122	"	"	Continuous	"
00010110	25				
01010110	125				
00010001	21	Decode Parallel Word	Decode Parallel Word	Ext 2, IOPLS	
00101001	51	Encode	Dig. Controlled Attenuator	Ext 1, Start	
00011001	31	Decode Parallel Word	"	Ext 2, IOPLS	
10011001	231	Dig. Controlled Attenuator	"	-	
10010010	222	"	Encode	Ext 1, Start	
10010001	221	"	Decode Parallel Word	Ext 2, IOPLS	

major consideration throughout the physical board design, and accounts for the multiple generation of logical variables at various physical locations.

A.4.2 External Data Operation

The 'external data' bit for each channel controls the portion of a nine pole, two way data selector 'switch'. This data selector is shown as U1, U2, U3, in Figure A4. The data selector allows one of two sets of data to be applied to the COMDAC digital inputs.

When the external data bit is logically 'high', data held in a latch are applied to the COMDAC for digital to analog conversion. This 9 bit latch is shown as U5 and part of U4 in Figure A5. The latch may be loaded via the Nova data bus lines DATA 7 through DATA 15 and associated inverters.

The latch feeding channel A is loaded by a DATO.A pulse AND-ed with the device select signal by gate G6. The channel B latch is loaded by a DATO.B pulse AND-ed with the device select signal by gate G7, as is shown in Figure A7. These latches are both effectively latched on the negative edge of their respective DATO pulses. For channel B as in Figure A7, the 9th bit is held by a positive-edge triggered 7474 flip-flop, so gate G8 has been used to invert the clock waveform. Including the gate used to filter the DATO.B line on the bus terminator card, this gives a total of 3 gate delays between DATO.B negative edge and the clocking edge for the 9th bit.

It should be noted that when the external data bit is high for a channel, then the operation of the COMDAC is determined. It is then independent of the state of the send/receive bit and the successive approximation logic. Only the timing of the analog output sample and hold need then be checked. This timing relative to the data latching may be checked via the 'sequence' flags, as is detailed later.

When the external data bit is zero, the circuits of Figures A2 and A4 revert to an implementation similar to the serial data transceiving converter shown in the DAC-76 applications notes.

A.4.3 Operation of the Converter Section

Much of these circuits is shown in Figures A2 and A4. The tri-state gating within each serial data transceiving converter is not the same as that shown in the DAC-76 applications note, but has a broadly similar function. This is detailed later.

Added within each converter circuit are six CMOS bilateral switches. Refer to Figure A2. When the channel is encoding, or is decoding serial data, the switches are normally set as follows :

SW1 and SW2 closed,

SW3 and SW4 open,

SW5 open and SW6 closed.

This gives normal converter operation. Switches SW1, SW2, SW3, and SW4 are used only for current steering. There is consequently no signal modulation due to their finite ON resistance.

The following switch settings are usual however, when external parallel data are to be decoded (external data bit logically 'high').

SW1 and SW2 open,

SW3 and SW4 closed,

SW5 open and SW6 closed.

This allows extended range (9 bit) decoding which includes normal (8 bit) decoding as required.

During encode operation, the converter input (point A) is fed from an input sample and hold circuit. This sample and hold (Ha2425) can be used to also provide an operational amplifier function. The input sample and hold circuit provides an overall gain of 4 to bipolar analog input signals of up to ten volts peak. The differential input facility has been preserved because millivolt differences in the input signal are significant.

The values of gain defining resistors were measured with a digital meter and resistors selected to give precise ratios are shown with suffixes.

No bias current error cancelling resistors are shown in the feedback loops of the unity gain followers because these are within the overall sample and hold feedback loops. These are, in any case, merely symbolic representations of the hold capacitor buffers which are inaccessible within the Ha2425 package.

A hold capacitor value of 1000pF was chosen. The encode conversion is complete well within 20 µs of the hold command. Hence the sample value held may be expected to change by less than 0,14 µV.

During decode operation, the output sample and hold is used to provide also the differential current to voltage conversion. The use of a sample and hold for this function limits the output voltage to above -5 volts,

if the compliance voltage specifications for the I_{OD} output are to be strictly met. Hence an output range of $\pm$ 5 volts was chosen.

The 1000 pF output hold capacitor is about the largest possible value which would allow a 50 kHz PCM sampling rate. This capacitor may appear small for infrasound work, but its charge may be refreshed by pulses on the IOPLS or EXT 2 lines, as long as the COMDAC digital inputs are held steady.

The output sample and holds initially exhibited intermittent instability. This was caused, of course, by the parasitic capacitance between point C and ground - as contributed mainly by the series diode and zener diode. The resulting one volt rough sine wave output at about 2,5 MHz fundamental was not directly audible. Hence the 22pF capacitors were added to shunt the feedback resistors, and are sufficient.

A.4.4 The Serial Data Tri-State Gating

Refer to Figure A4. The purpose of the tri-state gating network is to route the serial data appropriately depending on the combined send/receive states of the two channels. It is necessary to have a separate serial data bus for each of the channels, so that they may both encode simultaneously.

When a channel is encoding (send/receive bit logically high), it impresses its serial data onto its own serial data line, via an enabled gate G10. The encoding channel successive approximation register (SAR) collects the serial data from the serial data line of its own channel, via an enabled gate G11.

When a channel is decoding serial data, it feeds its successive approximation register input via an enabled gate G12. Gate G12 has collected this data from, say, the serial data line of the other, encoding, channel. In the decoding channel, gate G11 is disabled. Tri-state gating is convenient because the outputs of gates G11 and G12 can both directly feed the SAR.

Consider the case where one channel is encoding, and the other channel decoding the serial data. It should be noted that the delay between the encoding comparator output and the encoding SAR data input is equal to the delay between that comparator output and the decoding SAR data input. This is of great importance, for the following reason. When the encoder is encoding a sample value which is near the boundary between two encode quantum intervals, then the comparator output may oscillate extremely rapidly (20 nanosecond period, say) during one clock cycle of the conversion.

The two SAR's are clocked by a common clock. In order for both SAR's to clock in the same data under the above circumstances, the path delays to the two SAR's should ideally be exactly matched (and the receiving gate logic thresholds). These paths have been implemented with equal numbers of each type of gate.

A.4.5 The Protection Zeners

Refer to Figure A2. The series diode and zener diode were connected at point B and at point C as safeguards that the signal voltages at these points would not go below about - 6 volts. These were felt to be necessary because of the use of sample and holds on the COMDAC current outputs.

Assume, for example, that a 10 volt peak sine wave at the input is being sampled at the Nyquist rate and coincidentally at the peak values. Sample 1 may then appear as +5 volts at point A. At the end of the encode sequence, the I_{OE}^{+} output will be sinking about 2mA - and will try to continue to do so until the successive approximation register is started again. The input sample and hold may then be set to track the input signal in order to acquire the next sample. The voltage at point A may thus be around -5 volts with I_{EO}^{+} still attempting to sink 2 mA through 2,5 kilohm. This condition may last about 7 μs before the successive approximation register starts again. Now the absolute maximum rating for the voltage at I_{EO}^{+} is V- plus 8 volts, in this case -7 volts. Hence an unprotected scheme would be th_ unsatisfactory, even if the user always kept this input voltage within the legitimate limits.

However, the possibility of improper operation is an additional compelling reason for the zener protection. It is possible, under various circumstances, for the output of the sample and hold (point A) to go considerably more negative than -7 volts, say down to -14 volts. Although the precise internal circuit details of the COMDAC are not available, it is reasonable to expect catastrophic failure of the COMDAC under the above circumstances.

The CMOS switches SW1 and SW3 are also protected by the zeners. The switches have power supply voltages of +6½ volts and -6½volts, and the signal must stay within these limits.

The series diode and zener diode at point C keep the voltage at this point from going too far negative. They operate, for example, when the analog output is holding at -5 volts and I_{OD}^{+} is then required to sink 2mA.

A.4.6 Attenuator Operation

Refer to Figure A2. When a channel is to be used as a digitally controlled attenuator, the CMOS switches are set as follows :

SW1 and SW2 open,

SW3 and SW4 closed,

SW5 closed and SW6 open.

This is achieved by setting to logical 'ones' in the mode control word both the external data bit and the attenuator bit. The attenuator setting can then be latched in exactly the same way as external data is latched.

An effect of the logically high attenuator bit is to cause the output sample and hold for that channel to be kept in the sample state, i.e. to track the signal. This is shown in Figure A6. The attenuator then operates like the scheme in the DAC-76 applications note, by modulation of the COMDAC reference current.

The analog signal input to the attenuator must be kept within the power supply voltages to switch SW5, say $\pm$ 6 volts.

Switch SW5 isolates the COMDAC from inadvertent modulating effects during normal encode/decode operation. The 1,5 megohm resistor on the attenuator input is to protect SW5 from static electricity damage when this switch is open.

Switch SW6 is closed during normal encode/decode operation. This leaves about 20 kilohms from V_R on the COMDAC to ground, as is required to balance the input bias current offsets on the COMDAC reference amplifier.

During attenuator operation however, switch SW6 is open. This will cause some small DC offset on the attenuator output, but is unlikely to be important as the attenuator is A.C. coupled at input anyway. The opening of SW6 does, however, considerably extend the lower frequency range of the attenuator. The attenuator input coupling then has a 1/RC value of about 50 Hz.

The CMOS switches are all controlled from the mode control latch. The circuitry which interfaces this TTL latch to CMOS switching levels is shown in Figure A3. This figure also shows the circuits used to derive supply voltages suitable for the CMOS switches.

A.5 System Timing

The primary purpose of the system timing scheme is to ensure that the input and output sample and holds are caused to sample at suitable times relative to the start or finish of a conversion. A conceptual representation of the scheme is shown in Figure A6.

An encode conversion may be started by an external TTL compatible clock EXT 1, or by a Nova input/output START pulse addressed to the system device code (44 octal). Alternatively, if either channel is set to continuous conversion mode, a new conversion is initiated at the completion of a conversion as indicated by $\overline{CC}$ going logically low on the SAR.

Monostable T1 is fired at the start of an encode sequence. The input sample and holds of both channels are allowed to acquire and track the inputs for a period T1, about 7μs. If either channel is not encoding, then this will not affect it. It is noted that both encode channels sample at the same time instants. This is not expected to be a great limitation in general as the conversion time of less than 20 μs will probably allow redundant conversions if required. The $\overline{Q}$ output of monostable T1 is used to control the Ha 2425 sample and holds as these sample on a logically low control input.

As the input Ha 2425's are switched to hold mode, monostable T2 is triggered. The $\overline{Q}$ output of monostable T2 starts both the SAR's on the next conversion clock low to high transition. The conversion clock is quietened during the input sampling period T1, via the Q output of monostable T1. Hence the conversion clock and monostable T2 are forced locked with any arbitrary sampling frequency applied at EXT 1 or START. This is shown in Figure A12. The circuitry used to implement the input timing and conversion clock is shown in Figures A7 and A8.

Referring again to Figure A6, the output sampling gate is opened for time T4 at the end of an encode serial data decode sequence. Hence for many PCM modes it is not necessary to provide an additional stimulus for the receive sample and hold. However, when decoding parallel words, the precise timing of decoder outputs may be controlled by an external TTL compatible source EXT 2, or by a Nova IOPLS addressed to the system. These signals may also be used to periodically refresh the voltage held by the hold capacitor. The implementation is shown in Figure A8. Note that time T4 must be less than T1.

A.6 Software Readable Flags

Refer to Figure A9. Each channel has a sequence flag This is useful when using the channel to decode a series of parallel words. A set-reset (S-R) flip-flop is set each time a word is latched for decoding by the LATCH A (or B) signal. The S-R flip-flop is reset when the decode output is entered into the output sample and hold by an EXT 2 or IOPLS signal.

If two successive words are latched without an intervening EXT 2 or IOPLS signal, then the sequence flag goes high. The sequence flag is the D-type flip-flop. The flag will remain high until there occurs an EXT 2 or IOPLS signal followed by a latch signal.

The sequence flag does not check the absolute timing of the latch and EXT 2 or IOPLS signals. If an EXT 2 or IOPLS signal is followed by a latch signal 4 to 5 μs later, undetected faulty decoding may occur. This is because the output sample and hold samples for 5 μs (time T3) and the COMDAC requires $\frac{1}{2}$ μs to settle.

The usual Nova peripheral busy and done flags are associated with encoding channels. Refer to Figure A10. The busy flag is set only by a Nova START pulse. The busy flag is reset and the done flag set by the negative edge of the conversion complete $\overline{CC}$ signal.

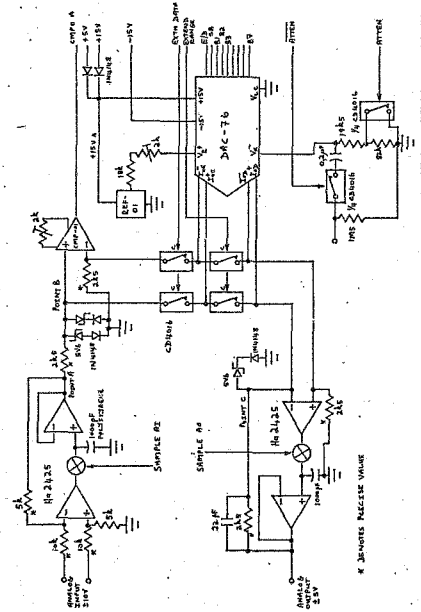


Figure A2 PCM System Analog Circuitry (Channel A).

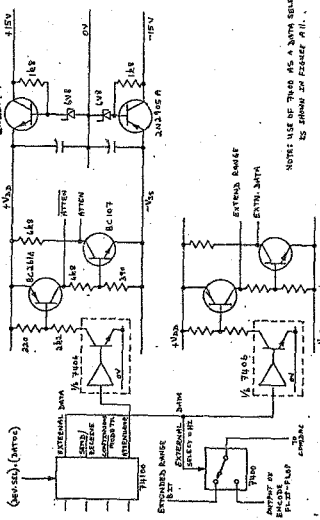


Figure A3 Analog Switch Control and TTL-CMOS Interface.

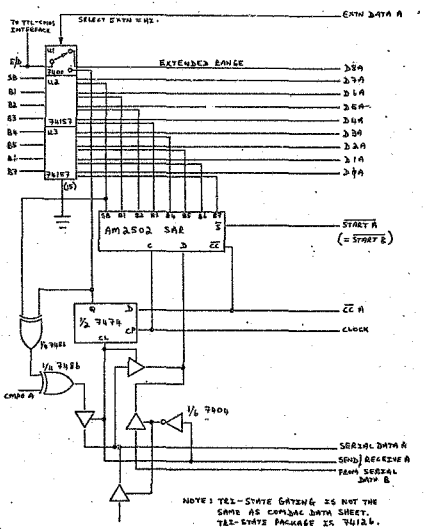


Figure A4 COMDAC Data Selector and Encode/Decode Logic.

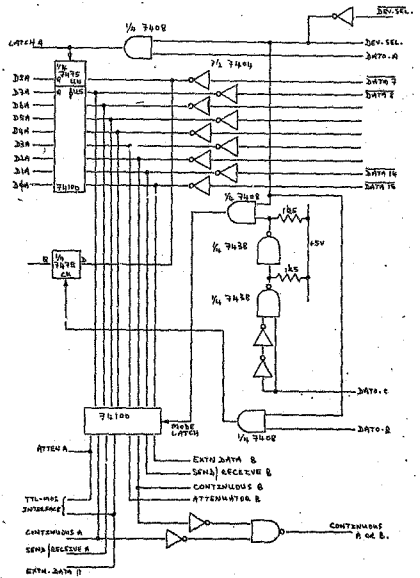


Figure A5 Mode Control Logic and (Channel A) Data Input.

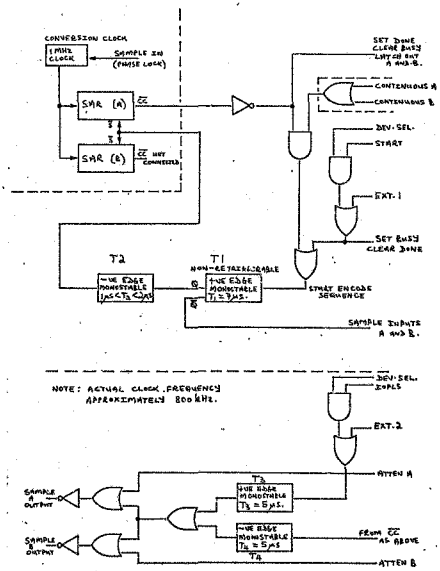


Figure A6 A Conceptual Representation of the PCM System Timing.

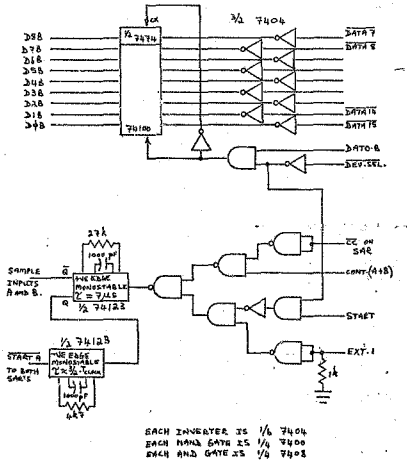


Figure A7 Timing and Control Logic for . out Sampling; Channel B Data Input.

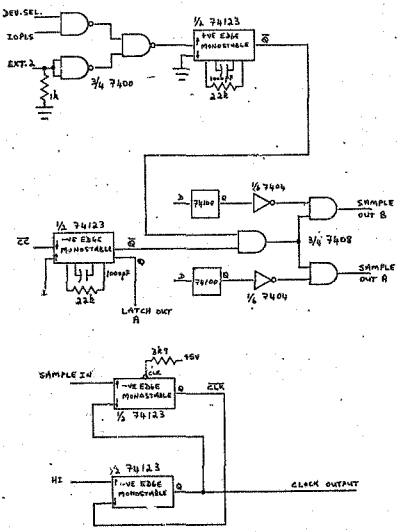


Figure A8 Timing and Control Logic for Output Sampling; Forced Look Conversion Clock.

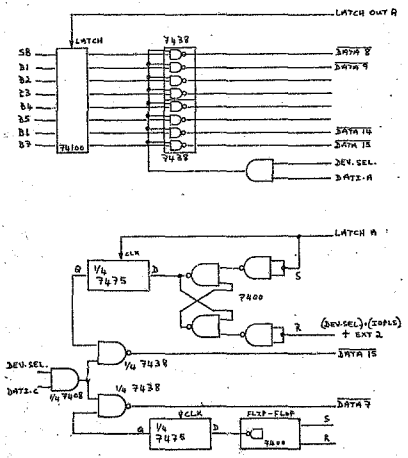


Figure A8 Parallel Data Output; Sequence Flags.

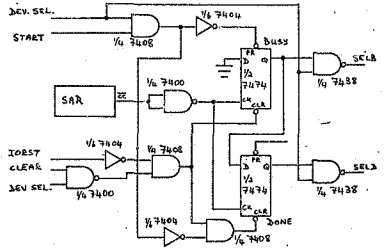


Figure A10 The Busy and Done Flags.

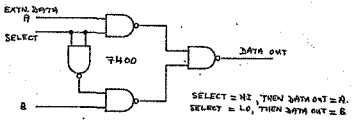


Figure A11 Details of Data Selector Using NAND Gates.

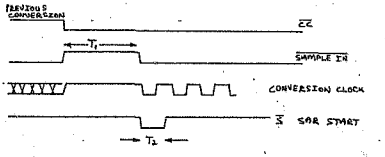


Figure A12 Timing Waveforms at Start of Encode Sequence.

APPENDIX B

A 12-BIT DIGITAL TO ANALOG CONVERTER FOR THE NOVA COMPUTER

B.1 Introduction

This unit provides a single channel sign plus 11 bit digital to analog conversion. It is a later inclusion in the acoustics instrumentation rack, and could not be assigned a single unique device code in the range 41_8 to 47_8. Consequently, it uses combinations of device code and Nova input/output control signal which are not used by the other devices in the rack.

The unit also incorporates a few more timing control elements than would be necessary for its operation alone. These elements allow, for example, co-ordination of the operation of the digital to analog converter with that of the experimental PCM system and the MP6912 analog to digital converter.

B.2 Circuit Description

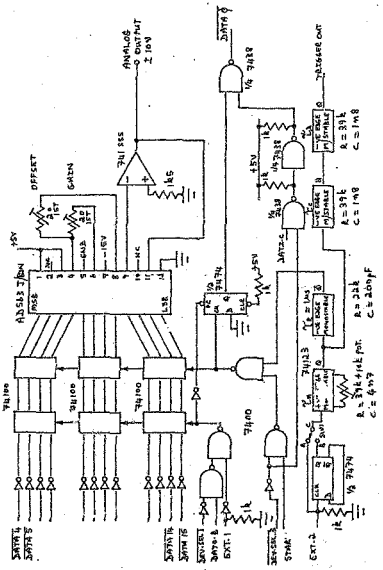
The circuitry is shown in Figure B1. Double buffering of the digital data is used. The first buffer may be loaded by a DATO.B signal from the Nova to device code one (41_8). This loads latches U1, U3, and U5 via data lines DATA 4 through DATA 15 and associated inverters. First buffer loading may also be initiated by EXT 1. This may be used, for example, to load the D/A directly from the MP6912 without using a Nova accumulator.

The contents of the input buffer may be transferred to the output buffer by a START signal from the Nova to device code 3. Alternatively, the transfer may be effected by an EXT 2 pulse. An adjustable delay is provided so that the buffer transfer occurs a time T_A after the positive edge of EXT 2. A TRIGGER OUT pulse is also available. This signal goes logically high at a time T_c after the output buffer has been updated, and remains high for time T_d. It may be used, for example, to trigger a fast sample and hold circuit if the settling time of the AD563 and 741SSS is too long.

By changing the jumpering of SW1 to BC, every second EXT 2 pulse may be used. One possible application of this is to drive EXT 1 from the $\overline{Q}$ output of U7. This will allow the converter to be run off a single external clock source.

Flip-flop U8 is a flag which may be read by a Nova DATI.C signal with device select 3. The flag is set when the first buffer is loaded. The flag is reset when the contents of the first buffer are transferred to the output buffer.

Considerable care should be taken in user routing of the analog output signal if 12-bit performance is to be realised. The output range is ±10 volts, giving an output step size of less than 5mV.


ANALOG OUTPUT Z 1004
OFFSET
GAIN
74LS55
AD565 JN
MSB
LSB
74LS74
74LS123
74LS04
74LS00
74LS10
74LS20
74LS30
74LS86
74LS08
74LS32
74LS27
74LS38
74LS51
74LS54
74LS86
74LS74
74LS00
74LS04
74LS10
74LS20
74LS30
74LS32

APPENDIX C A REVISION OF THE DUAL 10-BIT DIGITAL TO ANALOG CONVERTER FACILITY ON THE NOVA COMPUTER

C.1 Introduction

This board incorporates two D/A converters. Each channel gives a 9-bit plus sign analog output. There is also a third channel, and this has a one bit output. The board is a re-engineered version of a dual 10-bit D/A previously described in the Wits Acoustic Instrumentation Rack Laboratory Manual. Only the third channel circuitry introduced on the present board will be described.

C.2 Third Channel Circuit Description

The single bit output (Z) channel is programmed by two data bits. The Z channel may output a gated oscillator signal. The oscillator has a square wave output at TTL levels. The Z channel may also output a TTL level which does not oscillate.

Z value	Z output
ØØ (Binary)	TTL Low Level
Ø1	TTL High Level
1Ø	TTL High Level
11	TTL Oscillator output.

Alternatively, the Z channel may operate the pen lift on an H-P plotter. This is selected by plugging in the 8 pin DIL jumper select plug with a 180° rotation from its position for TTL outputs. The plug is shown as select switches SW1 and SW2 in Figure C2. Monostable T_1 allows the plotter to slow and settle before firing monostable T2 which drops the pen for a short time.

The oscillator is based on a 7413 Schmitt triggered NAND gate. The 7400 NAND gates form a single pole two-way data selector switch.

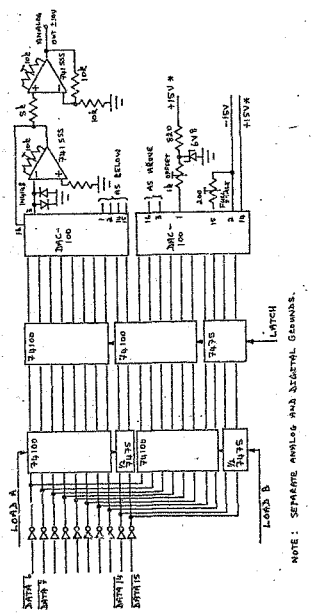
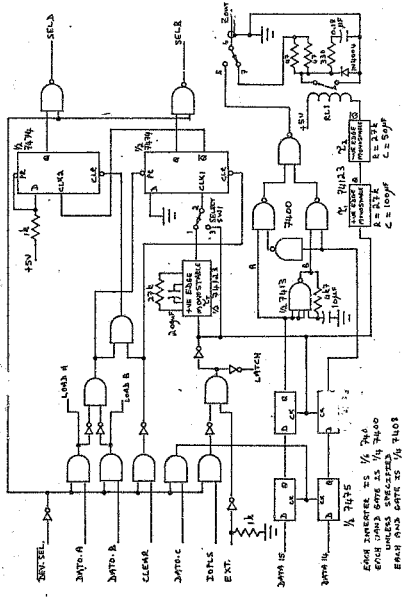


Figure C1 Buffered Latches and Digital t Analog Circuitry.


SEL.A
SEL.B
+5V
LOAD A
LOAD B
CLEAR
DATA.A
DATA.B
DATA.C
DATA.D
EXT.
DATA IS
DATA IS
74LS74
74LS00
74LS13
74LS123
74LS00
74LS00
74LS00

APPENDIX D SIMULATOR AND HARDWARE TESTING SOFTWARE FOR THE NOVA COMPUTER

```
C
C     D.1 PROBSPEC
C     ============
C
C     PROGRAM TO ENTER ARBITRARY TARGET PDF.
        INTEGER IE(512)
        DIMENSION E(512),X(101),XP(101)
        AREA=0.0
C       RGEN OUTPUT MAX 10 BITS ALLOWED FOR.
        DO 21 I=1,101
        X(I)=10.0
21      XP(I)=0.0
        DO 22 I=1,512
22      E(I)=0.0
        CALL FOPEN(1,"PDATA")
        WRITE(10,508)
508     FORMAT(/,'DO YOU WISH TO ENTER NEW SET PROB DENSIT
Y PTS?',
     $ /,' 1=YES',/)
        READ(11)NU
        IF(NU.NE.1) GO TO 29
30      WRITE(10,500)
500     FORMAT(/,' ENTER NO. OF EXPT PTS IN +VE HALF,TOTAL
 NO RANDOM
     $ VARIABLES REQD.',/)
        READ(11)N,M
        IF(M.GT.2**10) GO TO 30
        IF(N.GT.101) GO TO 30
        WRITE(10,501)N
501     FORMAT('  ENTER X(I)  XP(I) ON NEXT ',I3,' LINES.'
,//)
        DO 23 I=1,N
23      READ(11)X(I),XP(I)
        WRITE(10,502)
502     FORMAT(/,' ENTER REQD XMAX OF OUTPUT.',/)
        READ(11)XMAX
C       NOW BUILD UP DISK FILE "PDATA" .
        WRITE(1,503)N,M,XMAX
503     FORMAT(7X,I3,6X,I4,4X,F7.3)
        DO 32 I=1,N
32      WRITE(1,504)X(I),XP(I)
504     FORMAT(10X,F7.4,6X,F8.5)
        CALL FCLOS(1)
        GO TO 31
29      READ(1,509)N,M,XMAX
509     FORMAT(6X,I3,6X,I4,4X,F7.3)
        DO 33 I=1,N
33      READ(1,510)X(I),XP(I)
510     FORMAT(9X,F7.4,6X,F8.5)
        CALL FCLOS(1)
31      CONTINUE
C       INPUT DATA NOW READ.
C       CALCULATE DISCRETE PROBS. REQD.
C       DISTRIBUTION ASSUMED SYMMETRICAL.
```

```
      M=M/2
      D=XMAX/M
      KK=1
      KKI=KK+1
      DO 26 I=1,M
      II=I-1
      IF((II*D).GT.(X(KKI)+D/3.D))GO TO 5
4     SLOPE=(XP(KKI)-XP(KK))/(X(KKI)-X(KK))
      E(I)=(XP(KK)+SLOPE*(II*D-X(KK)))
      E(I)=E(I)*D
      AREA=AREA+E(I)
      GO TO 26
5     KK=KK+1
      KKI=KK+1
      GO TO 4
26    CONTINUE
      AREA=AREA-E(1)/2.D
      E(1)=E(1)/2.D
      WRITE(10,103)AREA
103   FORMAT('INITIAL AREA EQUALS  ',E12.4)
      AREAN=D.D
      FACTN=1.D/AREA
      DO 27 I=1,M
      E(I)=E(I)*FACTN
27    AREAN=AREAN+E(I)
      WRITE(10,106)AREAN
106   FORMAT('  NORMALISED AREA EQUALS   ',E12.4)
C     QUANTISE TO INTEGERS.
      AREA=D.DDD
      RANGE=32768.ODD
      DO 35 I=1,M
      E(I)=E(I)*RANGE
      K=E(I)
      K=K+1
      AREA=AREA+K
35    IE(I)=K
      AREA=AREA-RANGE
      K=AREA
      K=IABS(K)
      IF(K.LT.1) GO TO 37
      WRITE(10,511)AREA
511   FORMAT(/,' ERROR IN REQUIRED PROB SUMMATION=',F8
     .1,/,
     X' CORRECTION BY HAND? 1=YES. ',/)
      READ(11)L
      IF(L.EQ.1) GO TO 37
      L=K/3
      IA=AREA
      IS=IA/K
      IF(L.LT.1) GO TO 39
      DO 38 I=1,L
      IE(I)=IE(I)-3*IS
38    IA=IA-3*IS
39    L=L+1
      IA=IABS(IA)
      IF(IA.EQ.2) IE(L)=IE(L)-2*IS
      IF(IA.EQ.1) IE(L)=IE(L)-IS
C     SIGMA IE NOW EQUALS 32768
```

```
C       OUTPUT IE(I)
C       STORE REQD PROBS ON DISK FILE "RQPROB".
37      CALL FOPEN(2,"RQPROB")
        WRITE(2,505)M
505     FORMAT(11X,I4)
C       M MUST/SHOULD BE A MULTIPLE OF FOUR.
        M=M/4
        DO 24 J=1,M
        I=4*J-3
        IA=I+1
        IB=I+2
        IC=I+3
24      WRITE(2,506)IE(I),IE(IA),IE(IB),IE(IC)
506     FORMAT(1X,4(4X,I6))
        CALL FCLOS(2)
        WRITE(10,507)
507     FORMAT(//,'  PROBSP:I NOW COMPLETED. ',/)
        STOP
        END
C
C
C
C
C     D.2  ALICUT
C     ===========
C
        COMPILER DOUBLE PRECISION
C       PROGRAM TO FIND THE ALIAS AND CUTOFF VALUES
C       FROM THE REQUIRED PROBABILITIES.
        INTEGER B(1024),F(1024),C,D
        INTEGER A(1024),N,K,L
100     FORMAT(10X,I4)
101     FORMAT(4(4X,I6))
102     FORMAT(1X,4(4X,I6))
104     FORMAT(//,'  ALICUT:I COMPLETED. ',//)
        CALL FOPEN(1,"RQPROB")
        READ(1,100)N
C       N=NO. OF ABSOLUTE VALUES OF VARIABLES REQD.
C       NOW ENTER DESIRED PROBS.
        READ(1,101)(B(I),I=1,N)
        CALL FCLOS(1)
        RANGE=32767.0D0
        G=(RANGE+1.0D0)/N
        C=G
C       INITIALISE ARRAYS A,B,F
        DO 1 J=1,N
        A(J)=J
        F(J)=0
1       B(J)=B(J)-C
C       FIND THE LARGEST POSITIVE AND NEGATIVE DIFFERENCES
C       AND THEIR POSITIONS IN ARRAY B.
        DO 2 I=1,N
        C=0
        D=0
        DO 3 J=1,N
        IF(B(J).GT.C) GO TO 44
        C=B(J)
        K=J
```

```
      GO TO 3
44    IF(B(J).LT.D) GO TO 3
      D=B(J)
      L=J
3     CONTINUE
      SUM=0.0
      DO 45 J=1,N
45    SUM=SUM+IABS(B(J))
      IF((SUM).LT.(.1E0)) GO TO 6
C      ASSIGN THE ALIAS AND CUTOFF VALUES.
      A(K)=L
      F(K)=RANGE*B(L)*N+1
      B(K)=0.0
      B(L)=C+D
2     CONTINUE
C     COMPUTATION OF ALIAS AND CUTOFF VALUES COMPLETE.
107   FORMAT(11X,I4)
6     CONTINUE
      CALL FOPEN(3,"RNGDATA")
      WRITE(3,107)N
      N=N/4
      DO 4 J=1,N
      I=4*J-3
      IA=I+1
      IB=I+2
      IC=I+3
4     WRITE(3,102)A(I),A(IA),A(IB),A(IC)
      DO 5 J=1,N
      I=4*J-3
      IA=I+1
      IB=I+2
      IC=I+3
5     WRITE(3,102)F(I),F(IA),F(IB),F(IC)
      CALL FCLOS(3)
      WRITE(10,104)
      STOP
      END
C
C
C
C
C     D.3  RNGEN
C     ==========
C
      COMPILER DOUBLE PRECISION
C     GENERATE PSEUDO-RANDOM NOS. WITH DESIRED FREQ. DIS
TRIB.
      INTEGER A(512)
      INTEGER F(512),B(1024)
      CALL FOPEN(2,"RNGDATA")
      WRITE(10,110)
110   FORMAT(//,' NEW SET OF INITIAL CONDITIONS REQUIRED
? 1=YES,')
      READ(11)L
      IF(L.NE.1) GO TO 46
C     ENTER INITIAL CONDITIONS.
      WRITE(10,111)
```

```
111     FORMAT(/,'ENTER NO. OF INITIAL SAMPLES TO BE SKIPP
ED,NO. SAMPLES
 REQD')
        READ(11)L,M
        WRITE(10)L,M
        CALL FOPEN(1,"INITCOND")
        WRITE(1,112)L,M
        CALL FCLOS(1)
112     FORMAT(1X,I5,3X,I5)
46      CALL FOPEN(1,"INITCOND")
        READ(1,113)L,M
113     FORMAT(I5,3X,I5)
        CALL FCLOS(1)
        READ(2,114)N
114     FORMAT(10X,I4)
        READ(2,115)(A(I),I=1,N)
115     FORMAT(4(4X,I6))
        READ(2,115)(F(I),I=1,N)
        CALL FCLOS(2)
C       INPUT DATA ALL READ NOW.
        CALL FOPEN(1,"RANDNOS")
        WRITE BINARY (1)M,N
C       SET B TO ZERO.
        IX=N*2
        DO 9 I=1,IX
9       B(I)=0
C       LOOP TO GENERATE RANDOM NUMBERS.
        IX=33
        IXX=54
        IT=M*L
        DO 10 I=1,IT
        CALL RANDU(IX,IY,Y)
C       SIGN IS PASSED BACK AS IX.
        K=Y*N+1
        CALL RNGU(IXX,IYY)
        IXX=IYY
        IF(F(K).GT.IYY) GO TO 11
        K=A(K)
11      IF(I.LT.(L+1)) GO TO 10
        KS=(K-1)*IX
        IX=IY
        K=KS+N
        WRITE BINARY (1)KS
        B(K)=B(K)+1
10      CONTINUE
        CALL FCLOS(1)
        CALL FOPEN(2,"OCCURRED")
        K=0
        N=N*2
        WRITE(2,118)N,K
118     FORMAT(1X,I5,2X,I4)
        RANGE=32768.000
        RANGE=RANGE/M
        M=RANGE
        DO 47 I=1,N
47      B(I)=B(I)*M
C       B(I) ARE GENERATOR ACHIEVED PROBS.
        N=N/4
```

```
      DO 48 J=1,N
      I=4*J-3
      IA=I+1
      IB=I+2
      IC=I+3
48    WRITE(2,506)B(I),B(IA),B(IB),B(IC)
506   FORMAT(1X,4(4X,I6))
      CALL FCLOS(2)
      WRITE(10,120)
120   FORMAT(/,'  RNGEN:I NOW COMPLETED  ',/)
      STOP
      END
C
C
C
C
C       D.3.1  RNGSUB1
C       ==============
C
      COMPILER DOUBLE PRECISION
      SUBROUTINE RANDU(IX,IY,YFL)
      A=22039.0D0
      B=53.000
      Z=32768.0D0
      ZZ=2.0D0*Z
      YFL=DFLOAT(IX)
      YFL=YFL*A+B
      IF(YFL)1,2,2
1     YFL=-1.0D0*YFL
2     IX=2
      C=DMOD(YFL,ZZ)
      IF(C.GT.Z) IX=IX-Z
      IX=IX-1
      YFL=DMOD(YFL,Z)
      IY=YFL
      YFL=YFL/Z
      RETURN
      END
C
C
C
C
C       D.3.2  RNGSUB2
C       ==============
C
      A=22853.0D0
      B=16001.0D0
      Z=32768.0D0
      YFL=DFLOAT(IX)
      YFL=YFL*A+B
      IF(YFL)1,2,2
1     YFL=-1.0D0*YFL
2     YFL=DMOD(YFL,Z)
      IY=YFL
      RETURN
      END
C
C
```

```
C
C
C     D.4  FLTS
C     =========
C
      DOUBLE PRECISION C(10,5),X(10,5),Y(10,5),V
C     PROGRAM TO IMPLEMENT BANDPASS FILTERS (BUTTERWORTH
 PROTO'S).
1     WRITE(10,100)
100   FORMAT(/,' ENTER NUMBER OF POLES USED FOR L-P PRO
TOTYPE: ',/)
      READ(11)N
      IF((N/2*2).NE.N) GO TO 1
      IF(N.GT.20) GO TO 1
C     N USED TO SELECT DISK FILE OF CO-EFFICIENTS.
      IF(N.EQ.2) CALL FOPEN(1,"FLTD2")
      IF(N.EQ.4) CALL FOPEN(1,"FLTD4")
      IF(N.EQ.6) CALL FOPEN(1,"FLTD6")
      IF(N.EQ.8) CALL FOPEN(1,"FLTD8")
      IF(N.EQ.10) CALL FOPEN(1,"FLTD10")
      IF(N.EQ.12) CALL FOPEN(1,"FLTD12")
      IF(N.EQ.14) CALL FOPEN(1,"FLTD14")
      READ(1,101)NH
101   FORMAT(I4)
      IF(NH.NE.(N/2)) STOP BAD FILING.
      DO 2 I=1,NH
      DO 2 J=1,5
      READ(1,102)V
102   FORMAT(D25.18)
      C(I,J)=V
2     CONTINUE
      CALL FCLOS(1)
C     FILTER NOW DEFINED.
C     SET INITIAL CONDITIONS TO ZERO FOR FIRST PASS.
      DO 3 I=1,NH
      DO 3 J=1,5
      X(I,J)=0.000
      Y(I,J)=0.000
3     CONTINUE
C     NOW FIRST PASS THROUGH RANDNOS.
      IPASS=1
4     CALL FOPEN(1,"RANDNOS")
      CALL FOPEN(2,"FLTO")
      READ BINARY(1)M,NNN
      WRITE BINARY(2)M
      DO 8 J=1,M
      READ BINARY(1)IS
      IF((IPASS.EQ.1).AND.(J.LT.M-1000)) GO TO 8
C     SHIFT X AND Y ONE DELAY FOR EACH SECTION.
      DO 7 I=1,NH
      DO 5 K=2,5
      KI=7-K
      KK=KI-1
      X(I,KI)=X(I,KK)
      Y(I,KI)=Y(I,KK)
5     CONTINUE
C     NEW INPUT AND OUTPUT FOR THE SEGMENT:
      X(1,1)=IS
```

```
      IF(I.EQ.1) GO TO 6
      ID=I-1
      X(I,1)=Y(ID,1)
6     Y(I,1)=X(I,1)-2.0*X(I,3)+X(I,5)
      V=C(I,2)*Y(I,2)+C(I,3)*Y(I,3)+C(I,4)*Y(I,4)+C(I,5)
*Y(I,5)
      Y(I,1)=Y(I,1)-V
      Y(I,1)=Y(I,1)/C(I,1)
7     CONTINUE
      IF(IPASS.EQ.1) GO TO 8
      SY=Y(NR,1)
      WRITE BINARY(2)SY
8     CONTINUE
      CALL FCLOS(1)
      CALL FCLOS(2)
      IPASS=IPASS+1
      IF(IPASS.LT.3) GO TO 4
      STOP SEQUENCE FILTERED.
      END
C
C
C
C
C     D.5  FILTO
C     ==========
C
C     PROGRAM TO IMPLEMENT ALL-PASS FILTER
      CALL FOPEN(1,"RANDNOS")
      CALL FOPEN(2,"FLTO")
      READ BINARY(1)M,N
      WRITE BINARY (2)M
      DO 1 I=1,M
      READ BINARY (1)IS
      A=IS
1     WRITE BINARY (2)A
      CALL FCLOS(1)
      CALL FCLOS(2)
      STOP
      END
C
C
C
C
C     D.6  BUTTER
C     ===========
C
      COMPILER DOUBLE PRECISION
      COMPLEX A,B,U
      DIMENSION TH(20),C(5)
C     PROGRAM TO SET UP DIGITAL BANDPASS FILTERS.
C     USES EVEN ORDER BUTTERWORTH PROTOTYPES AND BI-LINE
AR XFM.
      WRITE(10,100)
100   FORMAT(/,' ENTER SAMPLING FREQ IN HZ. : ',/)
      READ(11)FS
      FQ=FS/2.000
      WRITE(10,101)
```

```
101     FORMAT(/,' ENTER LOWER, UPPER BAND EDGE FREQS IN H
Z. : ',/)
        READ(11)F1,F3
4       WRITE(10,102)
102     FORMAT(/,' ENTER REQD ORDER (EVEN) OF LOW-PASS PRO
TOTYPE. : ',/)
        READ(11)N
        N=N/2*2
        IF(N.GT.20) GO TO 4
C       TRANSFORM PROCEDURE :
        PI=4.0D0*DATAN(1.0D0)
        PI2=PI/2.0D0
        V1=F1/FO
        V3=F3/FO
        BB=V3-V1
        O=DTAN(V1*PI2)
        E=DTAN(V3*PI2)
        O=DSQRT(O*E)
        E=DATAN(O)
        V2=E/PI2
C       ASSUME UNITY LAMBDAR TO CALCULATE XFM CONSTANTS.
        O=DTAN(BB*PI2)
        O=1.0D0/O
        E=2.0D0*DCOS((V3+V1)*PI2)
        F=DCOS(BB*PI2)
        E=E/F
C       DECOMPOSE N-TH ORDER PROTO TO SECOND ORDER CASCADE
 SECTIONS.
C       THEN BI-LINEAR TRANSFORM EACH SECTION.
        ED=E*O
        EDS=ED**2
        OS=O**2
        O2=2.0*O
        N2=N*2
        NH=N/2
        DO 1 I=1,NH
1       TH(I)=PI2+PI/N2+(I-1)*PI/N
        CALL FOPEN(1,"FLTO")
        WRITE(1,103)NH
103     FORMAT(1X,I4)
        DO 3 J=1,NH
        X=-1.0D0*DCOS(TH(J))
        C(1)=OS+O2*X+1.0
        C(2)=-1.0*ED*(O2+2.0*X)
        C(3)=2.0*OS+EDS-2.0
        C(4)=-1.0*ED*(O2-2.0*X)
        C(5)=OS-O2*X+1.0
C       NORMALISE MID-BAND GAIN TO UNITY FOR THE SECTION.
        W=V2*PI
        X=DCOS(W)
        Y=DSIN(W)
        U=DCMPLX(X,Y)
        A=U**4-2.0*(U**2)+1.0
        B=C(1)*(U**4)+C(2)*(U**3)+C(3)*(U**2)+C(4)*U+C(5)
        A=A/B
        HM=DCABS(A)
        DO 2 I=1,5
        C(I)=C(I)*HM
```

```
2       WRITE(1,104)C(I)
104     FORMAT(1X,D25.18)
3       CONTINUE
        CALL FCLOS(1)
        STOP
        END
C
C
C
C
C     D.7  RESP
C     ========
C
        COMPILER DOUBLE PRECISION
C       PROGRAM TO CALCULATE RESPONSE OF BUTTERWORTH B-P F
ILTERS.
        COMPLEX Z,X,Y
        DIMENSION C(10,5),H(10)
1       WRITE(10,100)
100     FORMAT(/,'  ENTER NUMBER OF POLES USED FOR L-P PRO
TOTYPE: ',/)
        READ(11)N
        IF((N/2*2).NE.N) GO TO 1
        IF(N.GT.20) GO TO 1
C       N USED TO SELECT DISK FILE OF CO-EFFICIENTS.
        IF(N.EQ.2) CALL FOPEN(1,"FLTD2")
        IF(N.EQ.4) CALL FOPEN(1,"FLTD4")
        IF(N.EQ.6) CALL FOPEN(1,"FLTD6")
        IF(N.EQ.8) CALL FOPEN(1,"FLTD8")
        IF(N.EQ.10) CALL FOPEN(1,"FLTD10")
        IF(N.EQ.12) CALL FOPEN(1,"FLTD12")
        IF(N.EQ.14) CALL FOPEN(1,"FLTD14")
        READ(1,101)NH
101     FORMAT(I4)
        IF(NH.NE.(N/2)) STOP BAD FILING.
        DO 2 I=1,NH
        DO 2 J=1,5
        READ(1,102)Y
102     FORMAT(D25.18)
        C(I,J)=Y
2       CONTINUE
        CALL FCLOS(1)
C       FILTER NOW DEFINED.
        WRITE(10,103)
103     FORMAT(/,'  ENTER NUMBER OF PTS. OF EVALUATION OF
RESPONSE: ',/)
        READ(11)MP
        WRITE(10,560)
560     FORMAT(/,'  IS OUTPUT TO BE PLOTTED? 1=YES. : ',/)
        READ(11)IP
        IF(IP.NE.1) GO TO 3
        CALL FOPEN(2,"PLOT")
        WRITE(2,105)
105     FORMAT(' .RDX 10 ;SET RADIX TO DECIMAL. ')
        INCR=1024/MP
        WRITE(2,540)INCR
540     FORMAT('  .DEL: ',I3,' ;X-AXIS INCREMENT. ')
        WRITE(2,536)
```

```
536     FORMAT(' ,PNTR: PNTR ;STORES ADDRESS OF FIRST SAMP
LE.')
        WRITE(2,537)MP
537     FORMAT('  ,SIZE: ',I4,' ;NUMBER OF SAMPLES. ')
3       PI=4.0*DATAN(1.D)
C       FOR EACH PT. OF EVALUATION OF H(Z):
        DO 6 K=1,MP
        W=PI*(K-1)
        W=W/MP
        A=DCOS(W)
        B=DSIN(W)
        Z=DCMPLX(A,B)
C       FOR EACH SECTION OF H(Z):
        DO 4 I=1,NH
        X=Z**4-2.0*(Z**2)+1.D
        Y=C(I,1)*(Z**4)+C(I,2)*(Z**3)+C(I,3)*(Z**2)+C(I,4)
*(Z)+C(I,5)
        X=X/Y
        H(I)=DCABS(X)
4       CONTINUE
        AH=1.DDD
        DO 5 I=1,NH
5       AH=AH*H(I)
C       REQUIRE AMPLITUDE PLOT IN DB'S :
        IF(ABS(AH).LT.0.1E-15) AH=0.1E-15
        AH=20.0*DLOG10(AH)
C       SCALE FOR PLOT OF +20B TO -100DB :
        IAH=1000+10.0*AH
        IF(IAH.LT.0) IAH=0
        IF(IAH.GT.1023) IAH=1023
        VV=W/PI
        WRITE(10,561)VV,AH
561     FORMAT(/,'  V= ',F5.3,'   :     MOD(H)= ',F6.1,'DB .
 ')
        IF(IP.NE.1) GO TO 6
C       WRITE DISK FILE ENTRY FOR PLOTTING:
        IF(K.EQ.1) WRITE(2,538)IAH
        IF(K.EQ.1) GO TO 6
        WRITE(2,106)IAH
538     FORMAT('  PNTR: ',I4,'  ;FIRST DATUM TO BE PLOTTED
 ')
106     FORMAT(T2,I5)
6       CONTINUE
        IF(IP.EQ.1) CALL FCLOS(2)
        STOP
        END
C
C
C
C
C       D.8  QUANTO
C       ===========
C
C         PROGRAM TO PATCH "FLTO" TO D/A SAMPLES.
          INTEGER PR(1024)
          WRITE(10,110)
110       FORMAT(/,' TEST SEQ. REQD FOR "IDLPCM" ? 1=YES : '
,/)
```

```
      READ(11)IDL
      CALL FOPEN(1,"FLTO")
      READ BINARY (1)M
      SUM=0.0E0
      SQSUM=0.0E0
      DO 1 I=1,M
      READ BINARY (1)A
      SUM=SUM+A
1     SQSUM=SQSUM+A**2
      CALL FCLOS(1)
      SUM=SUM/M
      SQSUM=SQRT(SQSUM/M)
      WRITE(10,100)SUM,SQSUM
100   FORMAT(/,'  SEQUENCE MEAN= ',F9.3,'   :  RMS= ',F9
.3,/)
      WRITE(10,101)
101   FORMAT(/,'  TEST SIGNAL LEVEL TO BE "X" DB DOWN ON
 PEAK INPUT.',
     //,'     ENTER X : ',/)
      READ(11)X
C     MAX X JUST LESS THAN ANALOG ATTEN STEP SIZE.
C     SMALL X PREFERRED.
C     CALCULATE REQUIRED RMS VOLTAGE, AND SEQUENCE SCALI
NG FACTOR.
      PEAK=2047.5
      X=10.0** (X/20.0)
      P1=PEAK/X
      SCALE=P1/SQSUM
      WRITE(10,539)
539   FORMAT(/,'   ENTER NUMBER OF VARIATES TO BE LUMPED
 TOGETHER ',
     //,'     FOR PROB DISTRIBUTION P.OT: BETWEEN 4 &32 INC
LUSIVE. ',/)
      READ(11)NP
      DO 2 I=1,1024
2     PR(I)=0
      CALL FOPEN(2,"TEST")
      IF(IDL.NE.1) WRITE(2,102)
102   FORMAT('   ;PROGRAM TO TEST ENCODER A, INPUT R. ',
     //,'  .RDX 10   ;SET RADIX TO DECIMAL. ',/,'  ,PNTR: P
NTR ')
      CALL FOPEN(1,"FLTO")
      READ BINARY (1)M
      IF(IDL.NE.1) WRITE(2,533)M
533   FORMAT('  .SIZE= ',I4,'   ;NUMBER OF SAMPLES. ')
      IF(IDL.EQ.1) WRITE BINARY(2)M
      SUM=0.0
      SQSUM=0.0
      DO 3 I=1,M
      READ BINARY (1)A
C     SCALE FOR REQD SEQUENCE RMS VOLTAGE.
      A=A*SCALE
C     ROUND OFF TO NEAREST INTEGER.
      SI=+1.0
      IF(A.LT.(-0.5)) SI=-1.0
      A=A+0.5*SI
      K=A
```

```
C       PEAK CLIP SIGNAL AMPLITUDE TO WITHIN -2048 TO +204
7.
        IF(K.LT.(-2048)) K=-2048
        IF(K.GT.(+2047)) K=+2047
C       UPDATE TEST SEQUENCE STATISTICAL PARAMETERS.
        RK=K
        SUM=SUM+RK
        SQSUM=SQSUM+RK**2
C       DC SHIFT FOR SIGNALS WITHIN 0 TO 4095.
        K=K+2048
C       BUILD UP QUANTISED "OCCURRED" PROBABILITIES FOR 10
-BIT PLOT.
        KP=K/NP+1
        PR(KP)=PR(KP)+1
        IF(IDL.EQ.1) WRITE BINARY(2)K
        IF(IDL.EQ.1) GO TO 3
        IF(I.EQ.1) WRITE(2,534)K
        IF(I.EQ.1) GO TO 3
        IF(I.EQ.M) WRITE(2,535)K
        IF(I.EQ.M) GO TO 3
        WRITE(2,104)K
534     FORMAT('  PNTR: ',I4,'  ;FIRST SAMPLE DATA. ')
535     FORMAT('  LAST: ',I4,'  ;LAST SAMPLE DATUM. ')
104     FORMAT(T2,I4)
3       CONTINUE
        CALL FCLOS(1)
        CALL FCLOS(2)
C       WRITE NORMALISED OCCURRED PROBS TO FILE FOR PLOTTI
NG.
        CALL FOPEN(1,"PLOT")
        WRITE(1,105)
105     FORMAT('  .RDX 10 ;SET RADIX TO DECIMAL. ')
        INCR=NP/4
        WRITE(1,540)INCR
540     FORMAT('  .DEL: ',I3,' ; X-AXIS INCREMENT. ')
        WRITE(1,536)
536     FORMAT('  .PNTR: PNTR  ;STORES ADDRESS OF FIRST S
AMPLE. ')
        MP=4096/NP
        WRITE(1,537)MP
537     FORMAT('  .SIZE: ',I4,' ;NUMBER OF SAMPLES.')
        L=0
        LL=0
        DO 5 I=1,MP
5       IF(PR(I).GT.L) L=PR(I)
        DO 6 I=1,9
6       IF(L.GT.(2**I)) LL=2**(9-I)
        DO 4 I=1,MP
        PR(I)=PR(I)*LL
        IF(I.EQ.1) WRITE(1,538)PR(I)
        IF(I.EQ.1) GO TO 4
        WRITE(1,106)PR(I)
538     FORMAT('  PNTR: ',I4,'  ;FIRST DATUM TO BE PLOTTED
. ')
4       CONTINUE
106     FORMAT(T2,I5)
        CALL FCLOS(1)
        SUM=SUM/M
```

```
      SQSUM=SQRT(SQSUM/M)
      WRITE(10,107)SUM,SQSUM
107   FORMAT(/,'   TEST SEQUENCE MEAN= ',F9.3,' ;  RMS='
,F9.3,/)
      SUM=20*ALOG10(PEAK/SQSUM)
      WRITE(10,108)SUM
108   FORMAT(/,'   TEST SEQUENCE IS ',F9.3,' DB DOWN ON
PEAK INPUT VOL
TAGE')
      STOP
      END
C
C
C
C
C     D.9  TESTP
C     ==========
C
 .RDX 8 ;BACK TO OCTAL
 PCM=44 ;PCM DEVICE CODE
 DTA1=41 ;12BIT D/A DEVICE CODE
 DTA3=43 ;12BIT D/A DEVICE CODE
 START: STA 3,BASE ;SYSTEM TABLE BASE ADDRESS
 LDA 0,MODE ;
 DOC 0,PCM ;SET PCM MODE
 LDA 0,.SIZE ;
 STA 0,COUNT ;LOADS LOOP COUNTER
 LDA 3,.PNTR ;START ADDRESS OF SAMPLES
 SKPBN PCM ;SYNCHRONISE WITH PCM CLOCK
 JMP .-1 ;
 SKPDN PCM ;
 JMP .-1 ;
  ;PCM JUST DONE
 LOOP: LDA 1,0,3 ;LOAD OUTGOING SAMPLE FROM MEMORY
 DOB 1,DTA1 ;SEND SAMPLE TO D/A
 NIOS DTA3 ;LATCH SAMPLE INTO D/A
 SKPBN PCM ;PCM STARTED CONVERSION YET?
 JMP .-1 ;NOT YET STARTED
 SKPDN PCM ;CONVERSION HAS STARTED, IS IT DONE?
 JMP .-1 ;NOT YET DONE
 DIA 1,PCM ;CONVERSION DONE, READ IN ENCODED WORD
 STA 1,0,3 ;STORE SAMPLE
 INC 3,3 ;POINT TO NEXT SAMPLE LOCATION
 DSZ COUNT ;ALL SAMPLES DONE?
 JMP LOOP ;NO, DO NEXT SAMPLE
 JMP DISK ;ALL DONE, NOW IMAGE CORE TO DISK
 COUNT: 0 ;STORE LOOP COUNTER
 MODE: 46 ;PCM MODE  TO B
 BASE: 1 ;REMEMBER SYSTEM TABLE BASE ADDRESS
 DISK: LDA 0,BLOCK ;
 LDA 1,.BDES ;
 LDA 2,.FILE ;
 LDA 3,BASE ;
 JSR @0.CVS,3 ;CALL PRIMITIVE FUNCTION DRIVER
 0.IMG ;IMAGE FUNCTION
 JMP RETN ;CALL RETURN TO DOOS
 .BDES: BDES ;
 BDES: .SIZE ;
```

```
    LAST
    100000
    .FILE: FILE
   BLOCK: 1
    .TXTM 1
    FILE: .TXT ZDATAZ ;
    RETN: LDA 3,BASE ;
    JMP D.HOM,3 ;RETURN TO SYSTEM
    HALT ;NO RETURN
   .END START
C
C
C
C
C.....D.10 .PLOTP
C      ==========
C
   CENT: 100
    TENX: 10
    TENY: 10
  .RDX 8
  DTA=43 ;OCTAL DEVICE CODE.
  COUNT:1 ;STORAGE FOR LOOP COUNTER
  BASE: 1 ;STORAGE FOR SYSTEM BASE TABLE ADDRESS
  ZERO: 0 ;CONSTANT
  MAX: 1777 ;10 BITS ONES
  START: STA 3,BASE ;REMEMBER SYSTEM ADDRESS
  LDA 0. SIZE ;NUMBER OF SAMPLES TO PLOT
  STA 0,COUNT ;LOAD LOOP COUNTER
  LDA 0,.DEL ;LOADS X AXIS INCREMENT.
  LDA 3,.PNTR ;POINT TO SAMPLE WITH ACC 3
  LDA 1,ZERO ;FIRST X VALUE
  LOOP: LDA 2,0,3 ;LOAD Y VALUE
  DOA 1,DTA ;WRITE OUT X VALUE
  DOB 2,DTA ;WRITE OUT Y VALUE
  NIOP DTA ;IOPLS TO START PL.
  SKPDN DTA ;DEVICE BUSY STILL?
  JMP .-1 ;STILL BUSY
  ADD 0,1 ;NEXT X VALUE
  INC 3,3 ;POINT TO NEXT Y VALUE
  DSZ COUNT ;ALL SAMPLES PLOTTED?
  JMP LOOP ;NO, DO NEXT SAMPLE
  LDA 1,ZERO ;YES, DRAW AXES
  LDA 2,ZE
  DOA 1,D
  DOB 2,DT
  NIOP DTA ;
  SKPDN DTA ;
  JMP .-1 ;
  LDA 0,CENT ;
 XAXIS: ADD 0,1 ;STEP ACC 1 BY 100 DECIMAL
  DOA 1,DTA ;SET NEXT X VALUE
  NIOP DTA ;PLOT POINT
  SKPDN DTA ;FINISHED THAT PT. ?
  JMP .-1 ;NO
  DSZ TENX ;YES, ALL X AXIS DONE?
  JMP XAXIS ;NO
  LDA 1,ZERO ;
```

```
 DOA 1,DTA ;SET X TO ZERO
 YAXIS: ADD 0,1 ;STEP BY 100 DECIMAL
 DOB 1,DTA ;SET NEXT Y VALUE
 NIOP DTA ;PLOT PT.
 SKPDN DTA ;FINISHED?
 JMP .-1 ;NO
 DSZ TENY ;YES, ALL Y AXIS DONE?
 JMP YAXIS ;NO, DO NEXT PT.
  LDA 3,BASE  ;
 JMP @.HOM,3 ;
 HALT ;
 .END START
C
C
C
C
C     D.11  IDLPCM
C     ============
C
C       PROGRAM TO SIMULATE IDEAL COMDAC ENCODER.
        CALL FOPEN(1,"TEST")
        IA=0
        READ BINARY(1)M
        CALL FOPEN(2,"DATA")
C       WRITE DUMMY HEADER:
        DO 3 I=1,3
3       WRITE BINARY(2)IA
        WRITE BINARY(2)M
C       SIMULATE SETTING OF ANALOG ATTENUATER:
        WRITE(10,100)
100     FORMAT(/,' ENTER ANALOG ATTEN. SETTING TO BE SIMUL
ATED: (+VE):',
        READ(11)S
        S=ABS(S)
        S=10.0**(S/20.0)
        DO 4 J=1,M
        READ BINARY(1)IV
        IV=IV-2048
        ISGN=1
        IF(IV.LT.0) ISGN=-1
        IV=IABS(IV)
C       SCALE IV FOR IDEAL ENCODER INPUT RANGE:
        V=IV
        V=V*8287.0/2047.5
        V=V/S
C       SIMULATE RAMP A/D USING COMDAC:
        DO 1 IC=1,8
        ICC=IC-1
        DO 1 IS=1,16
        ISS=IS-1
        ID=(2**ICC)*(ISS+17)
        ID=2*ID
        ID=ID-33
        IA=ICC
        IB=ISS
        IF(ID.GE.V) GO TO 2
1       CONTINUE
C       PACK ENCODER OUTPUT WORD:
```

```
2       IV=IA*16+IB
        IF(ISGN.EQ.1) IV=IV+128
        WRITE BINARY(2)IV
4       CONTINUE
        CALL FCLOS(1)
        CALL FCLOS(2)
        STOP ENCODER OUTPUT IN DATA FILE.
        END
C
C
C
C
C     D.12  CONVERT
C     =============
C
C       PROGRAM TO READ IMAGE FILE "DATA" OF ENCODE COMPRE
SSED SAMPLES
C       OUTPUTS DECODED (EXPANDED) SAMPLES.
        CALL FOPEN(1,"DATA")
        CALL FOPEN(2,"SMPLSI")
C       THROW OUT HEADER OF "DATA" .
        DO 1 I=1,3
1       READ BINARY (1)ID
        N=512
C       READ NUMBER OF SAMPLES IN BLOCK
        READ BINARY (1)M
        ITE BINARY (2)M,N
C       FOR EACH SAMPLE!
        DO 2 I=1,M
        READ BINARY (1)ID
C       DETERMINE SIGN BIT
        IS=-1
        IF(ID.GT.127) IS=+1
C       NOW GET SIGNAL AMPLITUDE.
        IF(IS.EQ.(+1)) ID=ID-128
C       FIND CHORD AND STEP NUMBERS, THEN EXPAND.
        IST=MOD(ID,16)
        ICH=ID/16
        ID=(2**ICH)*(2*IST+33)-33
        ID=ID*IS
C       WRITE EXPANDED SAMPLE TO "SPLSI".
2       WRITE BINARY (2)ID
        CALL FCLOS(1)
        CALL FCLOS(2)
        STOP
        END
C
C
C
C
C
C     D.13  FLTRS
C     ===========
C
        DOUBLE PRECISION C(10,5),X(10,5),Y(10,5),V,SPWR
C       PROGRAM TO IMPLEMENT BANDPASS FILTERS (BUTTERWORTH
 PROTO'S).
C       THIS FILTER GIVES THE SIGNAL LEVEL.
```

```
1       WRITE(10,100)
100     FORMAT(/,'  ENTER NO. OF POLES FOR L-P PROTOTYPE (
SIGNAL FILTER)
     : ',/)
        READ(11)N
        IF((N/2*2).NE.N) GO TO 1
        IF(N.GT.20) GO TO 1
C       N USED TO SELECT DISK FILE OF CO-EFFICIENTS.
        IF(N.EQ.2) CALL FOPEN(1,"FLTD2")
        IF(N.EQ.4) CALL FOPEN(1,"FLTD4")
        IF(N.EQ.6) CALL FOPEN(1,"FLTD6")
        IF(N.EQ.8) CALL FOPEN(1,"FLTD8")
        IF(N.EQ.10) CALL FOPEN(1,"FLTD10")
        IF(N.EQ.12) CALL FOPEN(1,"FLTD12")
        IF(N.EQ.14) CALL FOPEN(1,"FLTD14")
        READ(1,101)NH
101     FORMAT(I4)
        IF(NH.NE.(N/2)) STOP BAD FILING.
        DO 2 I=1,NH
        DO 2 J=1,5
        READ(1,102)V
102     FORMAT(D25.18)
        C(I,J)=V
2       CONTINUE
        CALL FCLOS(1)
C       FILTER NOW DEFINED.
C       SET INITIAL CONDITIONS TO ZERO FOR FIRST PASS.
        DO 3 I=1,NH
        DO 3 J=1,5
        X(I,J)=0.000
        Y(I,J)=0.000
3       CONTINUE
C       NOW FIRST PASS THROUGH SMPLSI.
        IPASS=1
4       CALL FOPEN(1,"SMPLSI")
        READ BINARY(1)M,NNN
        SPWR=0.000
        DO 8 J=1,M
        READ BINARY(1)IS
        IF((IPASS.EQ.1).AND.(J.LT.M-1000)) GO TO 8
C       SHIFT X AND Y ONE DELAY FOR EACH SECTION.
        DO 7 I=1,NH
        DO 5 K=2,5
        KI=7-K
        KK=KI-1
        X(I,KI)=X(I,KK)
        Y(I,KI)=Y(I,KK)
5       CONTINUE
C       NEW INPUT AND OUTPUT FOR THE SEGMENT:
        X(I,1)=IS
        IF(I.EQ.1) GO TO 6
        ID=I-1
        X(I,1)=Y(ID,1)
6       Y(I,1)=X(I,1)-2.0*X(I,3)+X(I,5)
        V=C(I,2)*Y(I,2)+C(I,3)*Y(I,3)+C(I,4)*Y(I,4)+C(I,5)
*Y(I,5)
        Y(I,1)=Y(I,1)-V
        Y(I,1)=Y(I,1)/C(I,1)
```

```
7       CONTINUE
        SPWR=SPWR+Y(NH,1)**2
8       CONTINUE
        CALL FCLOS(1)
        IPASS=IPASS+1
        IF(IPASS.LT.3) GO TO 4
        SPWR=DSQRT(SPWR)
        CALL FOPEN(2,"SLVL")
        WRITE(2,103)SPWR
103     FORMAT(1X,D25.18)
        CALL FCLOS(2)
        STOP SEQUENCE FILTERED.
        END
C
C
C
C
C     D.14  FLTRN
C     ===========
C
        DOUBLE PRECISION C(10,5),X(10,5),Y(10,5),V,SPWR,NP
WR,SNR
C       PROGRAM TO IMPLEMENT BANDPASS FILTERS (BUTTERWORTH
 PROTO'S).
C       THIS FILTER GIVES THE NOISE LEVEL.
1       WRITE(10,100)
100     FORMAT(/,' ENTER NO. OF POLES FOR L-P PROTOTYPE (
NOISE FILTER)
1 ',/)
        READ(11)N
        IF((N/2*2).NE.N) GO TO 1
        IF(N.GT.20) GO TO 1
C       N USED TO SELECT DISK FILE OF CO-EFFICIENTS.
        IF(N.EQ.2) CALL FOPEN(1,"FLTD1")
        IF(N.EQ.4) CALL FOPEN(1,"FLTD3")
        IF(N.EQ.6) CALL FOPEN(1,"FLTD5")
        IF(N.EQ.8) CALL FOPEN(1,"FLTD7")
        IF(N.EQ.10) CALL FOPEN(1,"FLTD9")
        IF(N.EQ.12) CALL FOPEN(1,"FLTD11")
        IF(N.EQ.14) CALL FOPEN(1,"FLTD13")
        READ(1,101)NH
101     FORMAT(I4)
        IF(NH.NE.(N/2)) STOP BAD FILING.
        DO 2 I=1,NH
        DO 2 J=1,5
        READ(1,102)V
102     FORMAT(D25.18)
        C(I,J)=V
2       CONTINUE
        CALL FCLOS(1)
C       FILTER NOW DEFINED.
C       SET INITIAL CONDITIONS TO ZERO FOR FIRST PASS.
        DO 3 I=1,NH
        DO 3 J=1,5
        X(I,J)=0.0D0
        Y(I,J)=0.0D0
3       CONTINUE
C       NOW FIRST PASS THROUGH SMPLSI.
```

```
4           ... FOPEN(1,"SMPLSI")
            EAD BINARY(1)M,NNN
            SPWR=0.0D0
            DO 8 J=1,M
            READ BINARY(1)IS
            IF((IPASS.EQ.1).AND.(J.LT.M-1000)) GO TO 8
C           SHIFT X AND Y ONE DELAY FOR EACH SECTION.
            DO 7 I=1,NH
            DO 5 K=2,5
            KI=7-K
            KK=KI-1
            X(I,KI)=X(I,KK)
            Y(I,KI)=Y(I,KK)
5           CONTINUE
C           NEW INPUT AND OUTPUT FOR THE SEGMENT:
            X(1,1)=IS
            IF(I.EQ.1) GO TO 6
            ID=I-1
            X(I,1)=Y(ID,1)
6           Y(I,1)=X(I,1)-2.0*X(I,3)+X(I,5)
            V=C(I,2)*Y(I,2)+C(I,3)*Y(I,3)+C(I,4)*Y(I,4)+C(I,5)
*Y(I,5)
            Y(I,1)=Y(I,1)-V
            Y(I,1)=Y(I,1)/C(I,1)
7           CONTINUE
            SPWR=SPWR+Y(NH,1)**2
8           CONTINUE
            CALL FCLOS(1)
            IPASS=IPASS+1
            IF(IPASS.LT.3) GO TO 4
            NPWR=DSQRT(SPWR)
            CALL FOPEN(2,"SLVL")
            READ(2,103)SPWR
103         FORMAT(D25.18)
            CALL FCLOS(2)
            SNR=20.0D0*DLOG10(SPWR/NPWR)
            WRITE(10,104)SNR
104         FORMAT(/,' UNCORRECTED SNR = ',F8.3,'DB.',/)
            NPWR=NPWR*4000.0D0/(3400.0D0-860.0D0)
            SPWR=SPWR-NPWR*100.0D0/4000.0D0
            SNR=20.0D0*DLOG10(SPWR/NPWR)
            WRITE(10,105)SNR
105       FORMAT(/,'    CORRECTED SNR = ',F8.3,'DB. ',/)
            STOP BYE
            END
C
C
C
C
```

APPENDIX E SIMULATION AND UTILITY SOFTWARE FOR THE IBM370 COMPUTER

```
C
C
C
C
C     E.1  INIT
C     =========
C
      DIMENSION X(101),XP(101)
      READ(5,300)N,M,XMAX
300   format(3w)
      DO 1 I=1,N
1     READ(5,300)X(I),XP(I)
        WRITE(6,508)
508   FORMAT(/,'DO YOU WISH TO ENTER NEW SET PROB DENSIT
Y PTS?',/,'  1=YES: ',/)
        READ(9,100)NU
        IF(NU.NE.1) GO TO 31
30      WRITE(6,500)
500     FORMAT(/,' ENTER NO. OF EXPT PTS IN +VE HALF,TOTAL
 NO RANDOM VARIABLES REQD.',/)
        READ(9,200)N,M
        IF(M.GT.2*N-1) GO TO 30
        IF(N.GT.101) GO TO 30
        WRITE(6,501)N
501     FORMAT('  ENTER X(I)  XP(I) ON NEXT ',I3,' LINES.'
,//)
        DO 23 I=1,N
23      READ(9,200)X(I),XP(I)
100   FORMAT(Y)
200   FORMAT(2Y)
        WRITE(6,502)
502     FORMAT(/,' ENTER REQD XM      OUTPUT.',/)
        READ(9,100)XMAX
C       NOW BUILD UP DATA FILE
31      WRITE(7,503)N,M,XMAX
503     FORMAT(6X,I3,6X,I4,4X,F7.3.
        DO 32 I=1,N
32      WRITE(7,504)X(I),XP(I)
504     FORMAT(9X,F7.4,6X,F8.5)
       WRITE(6,120)
120   FORMAT(/,'  PRINTOUT TO BE A PRINTED PLOT OF PROB. D
ISTRIBUTION?',/,'   1=YES: ',/)
      READ(9,100)IPLO
      WRITE(7,121)IPLO
121   FORMAT(T5,I4)
      WRITE(6,702)
702   FORMAT(/,'  PRINTOUT OF REQUIRED PROBS.? 1=YES:',/)
      READ(9,100)IPRO
      WRITE(6,703)
703   FORMAT(/,'  PRINTOUT OF ALIAS & CUT-OFF VALUES REQUI
RED? 1=YES:',/)
      READ(9,100)IPRO1
      WRITE(7,701)IPRO,IPRO1
701   FORMAT(1X,I4,3X,I4)
      WRITE(6,113)
```

```
113   FORMAT(/,'  NOW SET-UP OF RANDOM NUMBER GENERATOR: '
,/)
      read(5,100)idum
      read(5,200)idum,idum2
      read(5,200)l,m
        WRITE(6,110)
110     FORMAT(//,' NEW SET OF INITIAL CONDITIONS REQUIRED
? 1=YES.')
        READ(9,100)LD
        IF(LD.NE.1) GO TO 46
C       ENTER INITIAL CONDITIONS.
        WRITE(6,111)
111     FORMAT(/,'ENTER NO. OF INITIAL SAMPLES TO BE SKIPP
ED,NO. SAMPLES REQUIRED: ',/)
      READ(9,200)L,M
46    WRITE(7,112)L,M
112   FORMAT(1X,I5,3X,I5)
      WRITE(6,114)
114   FORMAT(/,' PRINTOUT OF GENERATOR ACHIEVED PROBS. RE
QUIRED? 1=YES!',/)
      READ(9,100)IPRO2
      WRITE(6,115)
115   FORMAT(/,'  PRINTOUT OF TEST SIGNAL (FILTERED) PROBS
. REQUIRED? 1=YES:',/)
      READ(9,100)IPRO3
      WRITE(7,112)IPRO2,IPRO3
      WRITE(6,725)
725   FORMAT(/,'  NOW SET-UP THE SEND FILTER: ',/)
        WRITE(6,720)
720     FORMAT(/,' ENTER SAMPLING FREQ IN HZ. : ',/)
        READ(9,100)FS
        WRITE(6,101)
101     FORMAT(/,' ENTER LOWER, UPPER BAND EDGE FREQS IN H
Z. : ',/)
        READ(9,200)F1,F3
4       WRITE(6,102)
102     FORMAT(/,' ENTER REQD ORDER (EVEN) OF LOW-PASS PRO
TOTYPE. : ',/)
        READ(9,100)NLPS
      WRITE(6,712)
712   FORMAT(/,'  SIMULATION TO STOP WHEN SEND PROBS. FOUN
D? 1=YES: ',/)
      READ(9,100)ISTP
      IF(IPLO.EQ.1) ISTP=1
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
710   FORMAT(1X,F10.3,2(2X,F10.3))
      WRITE(6,726)
726   FORMAT(/,'  ENTER NUMBER OF BITS RESOLUTION OF D/A :
',/)
      READ(9,100)IBDA
      IF(IPLO.EQ.1) IBDA=6
      WRITE(6,727)
727   FORMAT(/,'  TEST SIGNAL LEVEL TO BE X DB DOWN ON PEA
K INPUT.',/,/,'  ENTER REQUIRED X (DB): ',/)
      READ(9,100)RX
      WRITE(7,728)IBDA,RX
728   FORMAT(T4,I4,3X,F10.3)
```

```
      IF(ISTP.EQ.1) STOP
      WRITE(6,730)
730   FORMAT(/,'  ENTER ANALOG ATTENUATOR SETTING TO BE SI
MULATED (DB):',/,'  ON SEND SIDE:')
      READ(9,100)S
      WRITE(7,731)S
731   FORMAT('S,F10.3)
      WRITE(6,735)
735   FORMAT(/,'  ENTER SETTING OF ATTEN BEFORE LIN A/D:')
      READ(9,100)SR
      WRITE(6,736)
736   FORMAT('  ENTER NO. BITS RESOLUTION OF LIN A/D:')
      READ(9,100)IBDAR
      WRITE(6,737)
737   FORMAT('  TEST SET TO BE RUN BACK-TO-BACK? 1=YES:')
      READ(9,100)IRX
      RX=0.0
      IF(IRX.EQ.1) RX=-2000.0
      WRITE(7,728)IBDAR,RX
      WRITE(7,731)SR
      WRITE(6,732)
732   FORMAT(/,'  NOW SET-UP SIGNAL FILTER ON RECEIVE SIDE
.',/,)
      WRITE(6,101)
      READ(9,200)F1,F3
      WRITE(6,102)
      READ(9,100)NLPS
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
      WRITE(6,734)
734   FORMAT(/,'  NOW SET-UP RECEIVE NOISE FILTER.',/)
      WRITE(6,101)
      READ(9,200)F1,F3
      WRITE(6,102)
      READ(9,200)NLPS
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
      STOP
      END
/DATA
/INC DATA
C
C
C
C
C     E.2  PSPEC
C     ==========
C
      INTEGER B(1024),A(512),F(512),C,D,N,K,L
        INTEGER IE(512)
        DIMENSION E(512),X(101),XP(101),AR(32500)
        AREA=0.0
C       RGEN OUTPUT MAX 10 BITS ALLOWED FOR.
        DO 21 I=1,101
        X(I)=10.0
21      XP(I)=0.0
        DO 22 I=1,512
      IE(I)=0
```

```
22      E(I)=0.0
29      READ(5,509)N,M,KMAX
509     FORMAT(6X,I3,6X,I4,4X,F7.3)
        DO 33 I=1,N
33      READ(5,510)X(I),XP(I)
510     FORMAT(9X,F7.4,6X,F8.5)
      READ(5,121;IPLO
121   FORMAT(T5,I4)
      READ(5,701)IPRO,IPRO1
701   FORMAT(1X,I4,3X,I4)
C       INPUT DATA NOW READ.
C       CALCULATE DISCRETE PROBS. REQD.
C       DISTRIBUTION ASSUMED SYMMETRICAL.
        M=M/2
        DR=XMAX/M
        KK=1
        KKI=KK+1
        DO 26 I=1,M
        II=I-1
        IF((II*DR).GT.(X(KKI)+DR/3.0))GO TO 5
4       SLOPE=(XP(KKI)-XP(KK))/(X(KKI)-X(KK))
        E(I)=(XP(KK)+SLOPE*(II*DR-X(KK)))
        E(I)=E(I)*DR
        AREA=AREA+E(I)
        GO TO 26
5       KK=KK+1
        KKI=KK+1
        GO TO 4
26      CONTINUE
        AREA=AREA-E(1)/2.0
        E(1)=E(1)/2.0
        IF(IPLO.NE.1) WRITE(6,103)AREA
103     FORMAT('INITIAL AREA EQUALS  ',E12.4)
        AREAN=0.0
        FACTN=1.0/AREA
        DO 27 I=1,M
        E(I)=E(I)*FACTN
27      AREAN=AREAN+E(I)
        IF(IPLO.NE.1) WRITE(6,106)AREAN
106     FORMAT('  NORMALISED AREA EQUALS   ',E12.4)
C       QUANTISE TO INTEGERS.
        AREA=0.0D0
        RANGE=32768.0D0
        DO 35 I=1,M
        E(I)=E(I)*RANGE
        K=E(I)
        K=K+1
        AREA=AREA+K
35      IE(I)=K
        AREA=AREA-RANGE
        K=AREA
        K=IABS(K)
        IF(K.LT.1) GO TO 37
        IF(IPLO.NE.1) WRITE(6,511)AREA
511     FORMAT(/,'  ERROR IN REQUIRED PROB SUMMATION= ',F8
.1,/)
        L=K/3
        IA=AREA
```

```
      IS=IA/K
      IF(L.LT.1) GO TO 39
      DO 38 I=1,L
      IE(I)=IE(I)-3*IS
38    IA=IA-3*IS
39    L=L+1
      IA=IABS(IA)
      IF(IA.EQ.2) IE(L)=IE(L)-2*IS
      IF(IA.EQ.1) IE(L)=IE(L)-IS
C     SIGMA IE NOW EQUALS 32768
C     OUTPUT IE(I) AS REQUIRED.
37    CONTINUE
      IF(IPRO.NE.1) GO TO 560
        IF(IPLO.NE.1) WRITE(6,505)M
505     FORMAT(11X,I4)
C       M MUST/SHOULD BE A MULTIPLE OF FOUR.
        M=M/4
        DO 24 J=1,M
        IG=4*J-3
        IA=IG+1
        IB=IG+2
        IC=IG+3
24      IF(IPLO.NE.1) WRITE(6,506)IE(IG),IE(IA),IE(IB),IE(
     _IC)
        M=M*4
506     FORMAT(1X,4(4X,I6))
560   CONTINUE
C       PROGRAM TO FIND THE ALIAS AND CUTOFF VALUES
C       FROM THE REQUIRED PROBABILITIES.
      N=M
102     FORMAT(1X,4(4X,I6))
104     FORMAT(//,' ALICUT:I COMPLETED. ',//)
C       N=NO. OF ABSOLUTE VALUES OF VARIABLES REQD.
        RANGE=32767.000
        C=(RANGE+1.000)/N
        C=C
C       INITIALISE ARRAYS A,B,F
        DO 1 J=1,N
        A(J)=J
        F(J)=0
1       B(J)=IE(J)-C
c     enter rms initial conditions.
      read(5,112)linit,minit
c     read other plot requirements.
      read(5,112)ipro2,ipro3
c     number of graphs to be plotted.
      if(ipro.ne.1)ipro=0
      if(ipro2.ne.1) ipro2=0
      if(ipro3.ne.1) ipro3=0
      rmsidl=ipro+ipro2+ipro3
      IF(IPLO.EQ.1.AND.IPLO.EQ.1) CALL PLOTFM(IE,N,RMSIDL)
C       FIND THE LARGEST POSITIVE AND NEGATIVE DIFFERENCES
C       AND THEIR POSITIONS IN ARRAY B.
        DO 2 I=1,N
        C=0
        D=0
        DO 3 J=1,N
        IF(B(J).GT.C) GO TO 44
```

```
      C=B(J)
        K=J
        GO TO 3
44      IF(B(J).LT.D) GO TO 3
        D=B(J)
        L=J
3       CONTINUE
        SUM=0.0
        DO 45 J=1,N
45      SUM=SUM+IABS(B(J))
        IF(SUM.LT.(.1E0)) GO TO 6
C       CALCULATE THE ALIAS AND CUTOFF VALUES.
        A(K)=L
        F(K)=RANGE+C*N+1
        B(K)=0.0
        B(L)=C+D
2       CONTINUE
C       COMPUTATION OF ALIAS AND CUTOFF VALUES COMPLETE.
6       CONTINUE
      IF(IPRO1.NE.1) GO TO 700
      IF(IPLO.EQ.1) GO TO 700
        N=N/4
        DO 704 J=1,N
        IG=4*J-3
        IA=IG+1
        IB=IG+2
        IC=IG+3
704     IF(IPLO.NE.1) WRITE(6,102)A(IG),A(IA),A(IB),A(IC)
        DO 705 J=1,N
        IG=4*J-3
        IA=IG+1
        IB=IG+2
        IC=IG+3
705     IF(IPLO.NE.1) WRITE(6,102)F(IG),F(IA),F(IB),F(IC)
      N=N*4
700   CONTINUE
        IF(IPLO.NE.1) WRITE(6,104)
C       GENERATE PSEUDO-RANDOM NOS. WITH DESIRED FREQ. DIS
TRIB.
C       ENTER INITIAL CONDITIONS.
112     FORMAT(1X,I5,3X,I5)
      l=linit
      m=minit
C       SET B TO ZERO.
        IX=N*2
        DO 9 I=1,IX
9       B(I)=0
C       LOOP TO GENERATE RANDOM NUMBERS.
        IX=33
        IXX=54
      Y=10.0
        IT=M+L
      MB=M
        DO 10 I=1,IT
        CALL RANDU(IX,IY,Y)
C       SIGN IS PASSED BACK AS IX.
        K=Y*N+1
        CALL RNGU(IXX,IYY)
```

```
      IXX=IYY
      IF(I(K).GT.IYY) GO TO 11
      K=A(K)
11    IF(I.LT.(L+1)) GO TO 10
      KS=(K-1)*IX
      IX=IY
      K=KS+N
     IJKL=J-L
     AR(IJKL)=KS
      B(K)=B(K)+1
10    CONTINUE
     IF(IPRO2.NE.1) GO TO 706
      K=0
      N=N*2
      RANGE=32768.000
      RANGE=RANGE/M
      M=RANGE
      DO 47 I=1,N
47    B(I)=B(I)*RANGE
C      B(I) ARE GENERATOR ACHIEVED PROBS.
     IF(IPLO.EQ.1) CALL PLFMT(B,N)
      N=N/4
      DO 48 J=1,N
      IG=4*J-3
      IA=IG+1
      IB=IG+2
      IC=IG+3
48    IF(IPLO.NE.1)WRITE(6,506)B(IG),B(IA),B(IB),B(IC)
     N=N*2
706   IF(IPLO.NE.1) WRITE(6,120)
120   FORMAT(/,'  RNGEN:1 NOW COMPLETED  ',/)
     READ(5,112)NLPS,ISTP
     READ(5,710)FS,F1,F3
710   FORMAT(1X,F10.3,2(2X,F10.3))
     ISW=1
     FPWR=0.0
     CALL FLTS(AR,MB,NLPS,FS,F1,F3,ISW,FPWR)
     IF(IPLO.NE.1) WRITE(6,711)
711   FORMAT(/,'  SEQUENCE FILTERED BY SEND FLTS. ',/)
     READ(5,728)IBDA,RX
     ARPK=2**(IBDA-1)-1.0+0.5
     CALL QUANTO(AR,MB,IBDA,RX)
     IF(IPLO.NE.1) WRITE(6,729)
728   FORMAT(T4,I4,3X,F10.3)
729   FORMAT(/,'  QUANTO COMPLETED. ',/)
     IF(IPLO.EQ.1.AND.IPRO3.EQ.1) CALL PRBPLT(AR,MB)
     IF(ISTP.EQ.1) STOP
     READ(5,730)S
     READ(5,728)IBDAR,RX
C    IBDAR IS LIN. A/D RESOLUTION (BITS).
C    RX.LT.(-1000) SHOWS TEST SET BACK-TO-BACK.
     IBTB=0
     IF(RX.LT.(-1000)) IBTB=1
     IF(IBTB.EQ.1) GO TO 561
     CALL IDLPCM(AR,MB,IBDA,S)
     IF(IPLO.NE.1) WRITE(6,731)
730   FORMAT(T5,F10.3)
731   FORMAT(/,'  IDLPCM COMPLETED.',/)
```

```
      CALL CONVRT(AR,MB)
      IF(IPLO.NE.1) WRITE(6,732)
732   FORMAT(/,' CONVERT COMPLETED.',/)
      S=0.0
      ARPK=8159.0
561   READ(5,730)SR
C     SR IS ANALOG ATTEN BEFORE LIN. A/D.
      SR=SR+S
      CALL ADCON(AR,MB,IBDAR,SR,ARPK)
      READ(5,112)NLPR,ISTP
      READ(5,710)FS,F1,F3
      F1S=F1
      F3S=F3
      ISW=0
      CALL FLTS(AR,MB,NLPR,FS,F1,F3,ISW,FPWR)
      SPWR=FPWR
      IF(IPLO.NE.1) WRITE(6,733)
733   FORMAT(/,' REC. SIGNAL FILTER COMPLETED.',/)
      READ(5,112)NLPR,ISTP
      READ(5,710)FS,F1,F3
      CALL FLTS(AR,MB,NLPR,FS,F1,F3,ISW,FPWR)
      QPWR=FPWR
      SNR=20.0*ALOG10(SPWR/QPWR)
      QPWR=QPWR*(FS/2.0)/(F3-F1)
      SPWR=SPWR-QPWR*(F3S-F1S)/(FS/2.0)
      SNRC=20.0*ALOG10(SPWR/QPWR)
      IF(IPLO.NE.1) WRITE(6,740)SNR,SNRC
740   FORMAT(/,'  SNR= ',F8.3,T25,' SNRC= ',F8.3,' (DB).',
     /)
        STOP
        END
      SUBROUTINE RANDU(IX,IY,Y)
      DOUBLE PRECISION YFLD,ZZ,Z,A,B,YFL,G,H
      A=22159.0DD
      B=6913.0DD
      G=58653.0DD
      H=13849.0DD
        Z=32768.0DD
        ZZ=2.0DD*Z
        YFL=DFLOAT(IX)
        YFL=YFL*A+B
        IF(YFL)1,2,2
1       YFL=-1.0DD*YFL
2     CONTINUE
C     GENERATE SIGN BIT,PASSED BACK AS IX:
      IF(Y.GT.2.0) YFLD=13.0DD
      YFLD=YFLD*G+H
      YFLD=DMOD(YFLD,ZZ)
      IY=YFLD
      IX=-1
      IF((IY/2)*2.EQ.IY) IX=+1
      YFL=DMOD(YFL,Z)
        IY=YFL
        Y=YFL/Z
        RETURN
        END
      SUBROUTINE RNGU(IX,IY)
      DOUBLE PRECISION YFL,Z,A,B
```

```
      A=27237.0D0
      B=6925.0D0
        Z=32768.0D0
        YFL=DFLOAT(IX)
        YFL=YFL*A+B
        IF(YFL)1,2,2
1       YFL=-1.0D0*YFL
2       YFL=DMOD(YFL,Z)
        IY=YFL
        RETURN
        END
      SUBROUTINE FLTS(AR,M,NLPS,FS,F1,F3,ISW,FPWR)
        DOUBLE PRECISION C(10,5),X(10,5),Y(10,5),V
      DIMENSION AR(32500)
C       PROGRAM TO IMPLEMENT BANDPASS FILTERS (BUTTERWORTH
     PROTO'S).
      NH=NLPS/2
      CALL BUTTER(NLPS,C,FS,F1,F3)
C       FILTER NOW DEFINED.
C       SET INITIAL CONDITIONS TO ZERO FOR FIRST PASS.
        DO 3 I=1,NH
        DO 3 J=1,5
        X(I,J)=0.0D0
        Y(I,J)=0.0D0
3       CONTINUE
C       NOW FIRST PASS THROUGH RANDNOS.
        IPASS=1
4     CONTINUE
      SQ=0.0
        DO 8 J=1,M
      IS=AR(J)
        IF((IPASS.EQ.1).AND.(J.LT.M-800D)) GO TO 8
C       SHIFT X AND Y ONE DELAY FOR EACH SECTION.
        DO 7 I=1,NH
        DO 5 K=2,5
        KI=7-K
        KK=KI-1
        X(I,KI)=X(I,KK)
        Y(I,KI)=Y(I,KK)
5       CONTINUE
C       NEW INPUT AND OUTPUT FOR THE SEGMENT:
        X(I,1)=IS
        IF(I.EQ.1) GO TO 6
        ID=I-1
        X(I,1)=Y(ID,1)
6       Y(I,1)=X(I,1)-2.0D0*X(I,3)+X(I,5)
        V=C(I,2)*Y(I,2)+C(I,3)*Y(I,3)+C(I,4)*Y(I,4)+C(I,5)
     *Y(I,5)
        Y(I,1)=Y(I,1)-V
        Y(I,1)=Y(I,1)/C(I,1)
7       CONTINUE
        IF(IPASS.EQ.1) GO TO 8
      SQ=SQ+Y(NH,1)**2
        IF(ISW.EQ.1) AR(J)=Y(NH,1)
8       CONTINUE
        IPASS=IPASS+1
        IF(IPASS.LT.3) GO TO 4
      FPWR=SQRT(SQ/M)
```

```
      RETURN
        END
      SUBROUTINE BUTTER(N,C,FS,F1,F3)
      IMPLICIT REAL*8 (A-E,G,H,O-Z,$)
      DOUBLE PRECISION C(10,5)
        COMPLEX*16 A,B,U
        DIMENSION TH(20)
C       PROGRAM TO SET UP DIGITAL BANDPASS FILTERS.
C       USES EVEN ORDER BUTTERWORTH PROTOTYPES AND BI-LINE
AR XFM.
        FO=FS/2.0
        N=N/2*2
C       TRANSFORM PROCEDURE :
        PI=4.0D0*DATAN(1.0D0)
        PI2=PI/2.0D0
        V1=F1/FO
        V3=F3/FO
        BB=V3-V1
        D=DTAN(V1*PI2)
        E=DTAN(V3*PI2)
        D=DSQRT(D*E)
        E=DATAN(D)
        V2=E/PI2
C       ASSUME UNITY LAMBDAR TO CALCULATE XFM CONSTANTS.
        D=DTAN(BB*PI2)
        D=1.0D0/D
        E=2.0D0*DCOS((V3+V1)*PI2)
        F=DCOS(BB*PI2)
        E=E/F
C       DECOMPOSE N-TH ORDER PROTO TO SECOND ORDER CASCADE
 SECTIONS.
C       THEN BI-LINEAR TRANSFORM EACH SECTION.
        ED=E*D
        EDS=ED**2
        DS=D**2
        D2=2.D*D
        N2=N*2
        NH=N/2
        DO 1 I=1,NH
1       TH(I)=PI2*PI/N2+(I-1)*PI/N
        DO 3 J=1,NH
        X=-1.0D0*DCOS(TH(J))
        C(J,1)=DS+D2*X+1.0
        C(J,2)=-1.0*ED*(D2+2.0*X)
        C(J,3)=2.0*DS+EDS-2.0
        C(J,4)=-1.0*ED*(D2-2.0*X)
        C(J,5)=DS-D2*X+1.0
C       NORMALISE MID-BAND GAIN TO UNITY FOR THE SECTION.
        W=V2*PI
        X=DCOS(W)
        Y=DSIN(W)
        U=DCMPLX(X,Y)
        A=U*U*U-(2.0D0,0.0D0)*(U*U)+(1.0D0,0.0D0)
        B=C(J,1)*(U*U*U*U)+C(J,2)*(U*U*U)+C(J,3)*(U*U)+C(J
,4)*U+C(J,5)
        A=A/B
        HM=CDABS(A)
        DO 2 I=1,5
```

```
        C(J,I)=C(J,I)*HM
2       CONTINUE
3         CONTINUE
        RETURN
          END
        SUBROUTINE QUANTO(AR,M,IBDA,X)
        DIMENSION AR(3Z500)
C         PROGRAM TO PATCH 'FLTO" TO D/A SAMPLES.
          SUM=0.0E0
          SQSUM=0.0E0
          DO 1 I=1,M
        A=AR(I)
          SUM=SUM+A
1         SQSUM=SQSUM+A**2
          SUM=SUM/M
          SQSUM=SQRT(SQSUM/M)
        IPLO=0
          IF(IPLO.NE.1) WRITE(6,10')SUM,SQSUM
100       FORMAT(/,'  SEQUENCE MEAN= ',F9.3,'   :  RMS= ',F9
.3,/)
C         MAX X JUST LESS THAN ANALOG ATTEN STEP SIZE.
C         SMALL X PREFERRED.
C         CALCULATE REQUIRED RMS VOLTAGE, AND SEQUENCE SCALI
NG FACTOR.
        IPEAK=2**(IBDA-1)
        PEAK=IPEAK
        PEAK=PEAK-0.5
          X=10.0**(X/20.0)
          P1=PEAK/X
          SCALE=P1/SQSUM
          SUM=0.0
          SQSUM=0.0
          DO 3 I=1,M
        A=AR(I)
C         SCALE FOR REQD SEQUENCE RMS VOLTAGE.
          A=A*SCALE
C         ROUND OFF TO NEAREST INTEGER.
          SI=+1.0
          IF(A.LT.(-0.5)) SI=-1.0
          A=A+0.5*SI
          K=A
C         PEAK CLIP SIGNAL AMPLITUDE TO WITHIN -2048 TO +204
7 (FOR 12-BIT D/A).
          IF(K.LT.(-IPEAK)) K=-IPEAK
          IF(K.GT.(+(IPEAK-1))) K=+(IPEAK-1)
C         UPDATE TEST SEQUENCE STATISTICAL PARAMETERS.
          RK=K
          SUM=SUM+RK
          SQSUM=SQSUM+RK**2
        AR(I)=K
3         CONTINUE
          SUM=SUM/M
          SQSUM=SQRT(SQSUM/M)
          IF(IPLO.NE.1) WRITE(6,107)SUM,SQSUM
107       FORMAT(/,'   TEST SEQUENCE MEAN= ',F9.3,'  :  RMS='
,F9.3,/)
          SUM=20*ALOG10(PEAK/SQSUM)
          IF(IPLO.NE.1) WRITE(6,108)SUM
```

```
108     FORMAT(/,'   TEST SEQUENCE IS ',F9.3,' DB DOWN ON
PEAK INPUT VOLTAGE. ',/)
        RETURN
        END
      SUBROUTINE IDLPCM(AR,M,IBDA,S)
      DIMENSION AR(32500)
C       PROGRAM TO SIMULATE IDEAL COMDAC ENCODER.
C       SIMULATE SETTING OF ANALOG ATTENUATER:
        S=10.0**(S/20.0)
        DO 4 J=1,M
      IV=AR(J)
        ISGN=1
        IF(IV.LT.0) ISGN=-1
        IV=IABS(IV)
C       SCALE IV FOR IDEAL ENCODER INPUT RANGE:
        V=IV
        V=V*8159.0/(2**(IBDA-1)-0.5)
        V=V/S
C       SIMULATE RAMP A/D USING COMDAC:
        DO 1 IC=1,8
        ICC=IC-1
        DO 1 IS=1,16
        ISS=IS-1
        ID=(2**ICC)*(ISS+17)
        ID=2*ID
        ID=ID-33
        IA=ICC
        IB=ISS
        IF(ID.GE.V) GO TO 2
1       CONTINUE
C       PACK ENCODER OUTPUT WORD:
2       IV=IA*16+IB
        IF(ISGN.EQ.1) IV=IV+128
      AR(J)=IV
4       CONTINUE
      RETURN
        END
      SUBROUTINE CONVRT(AR,M)
      DIMENSION AR(32500)
C       PROGRAM TO READ IMAGE FILE "DATA" OF ENCODE COMPRE
SSED SAMPLES
C       OUTPUTS DECODED (EXPANDED) SAMPLES.
C       FOR EACH SAMPLE:
        DO 2 I=1,M
      ID=AR(I)
C       DETERMINE SIGN BIT
        IS=-1
        IF(ID.GT.127) IS=+1
C       NOW GET SIGNAL AMPLITUDE.
        IF(IS.EQ.(+1)) ID=ID-128
C       FIND CHORD AND STEP NUMBERS, THEN EXPAND.
        IST=MOD(ID,16)
        ICH=ID/16
        ID=(2**ICH)*(2*IST+33)-33
        ID=ID*18
2     AR(I)=ID
      RETURN
        END
```

```
      SUBROUTINE PLOTFM(IE,N,RMSIDL)
      double precision rms,wv
      INTEGER IE(512)
      NREM=N
      iplo=rmsidl+0.5
c     calculate rms=standard deviation of required sequenc
e.
      rms=0.0d0
      do 1 i=1,n
      wv=i-1
      wv=(wv**2)*ie(i)
1     rms=rms+wv
      rms=dsqrt(rms/32768.0d0)
c     normalise rms per unit plot.
      rmsidl=rms/(2.0*n-1)
      rmsidl=20.0*alog10(1.0/rmsidl)
C     PREPARE IE(I) FOR PLOTTING.
C     FIRST CREATE WORKING SPACE IN ARRAY IE.
      IF(N.LT.512) GO TO 83
      N=N/2
      DO 84 I=1,N
84    IE(I)=IE(I*2-1)+IE(I*2)
83    CONTINUE
C     NOW SHIFT ASYMMETRIC DISTRIBUTION TO RIGHT.
      DO 85 I=1,N
85    IE(N+I)=IE(I)
C     NOW FOLD RIGHT HALF INTO LEFT HALF.
      DO 86 I=1,N
86    IE(N-I+1)=IE(N+I)
      N=N*2
C     NOW PACK AVERAGED IE INTO FIRST 64 LOCATIONS.
      IF(N.LT.64) GO TO 80
      ILP=N/64
      DO 81 I=1,64
      IX=(I-1)*ILP+1
      IY=0
      DO 82 J=1,ILP
      K=IX+J-1
82    IY=IY+IE(K)
81    IE(I)=IY
80    N=NREM
      write(7,100)iplo,rmsidl
100   format(t4,i3,t13,'RQPROB X=',f6.3)
C     SIGMA IE WAS 32768 AT START; CORRECT AREA UNDER PDF:
      DO 88 I=1,64
      R=IE(I)
88    IE(I)=R*0.5+0.5
      DO 87 I=1,16
87    WRITE(7,506)IE(I),IE(I+16),IE(I+32),IE(I+48)
506   FORMAT(1X,4(4X,I6))
      RETURN
      END
      SUBROUTINE PLFMI(B,N)
      INTEGER B(1024)
C     PACK B FOR 64 VALUES TO BE PLOTTED.
      IF(N.LE.64) GO TO 80
      ILP=N/64
      DO 81 I=1,64
      IX=(I-1)*ILP+1
```

```
      IY=0
      DO 82 J=1,ILP
      K=IX+J-1
82    IY=IY+B(K)
81    B(I)=IY
80    CONTINUE
      DO 87 I=1,16
87    WRITE(7,506)B(I),B(I+16),B(I+32),B(I+48)
506   FORMAT(1X,4(4X,I6))
      RETURN
      END
      SUBROUTINE PRBPLT(AR,MB)
      DIMENSION AR(32500)
      INTEGER B(64)
      DO 1 I=1,64
1     B(I)=0
C     FIND FILTERED SEQUENCE PROB DISTRIBUTION.
C     AR ELEMENTS INTEGERS BETWEEN -32 AND +31.
      DO 2 I=1,MB
      K=AR(I)
      K=K+33
2     B(K)=B(K)+1
C     SCALE TO NORMALISED INTEGER PROBS.
      DO 4 I=1,64
      R=B(I)
      R=R*32768.0/MB
4     B(I)=R
      DO 3 I=1,16
3     WRITE(7,506)B(I),B(I+16),B(I+32),B(I+48)
506   FORMAT(1X,4(4X,I6))
      RETURN
      END
      SUBROUTINE ADCON(AR,MB,N,S,ARPK)
      DIMENSION AR(32500)
C     ROUTINE TO SIMULATE N-BIT A/D CONVERTER.
C     ANALOG ATTENUATION OF S-DB REQUIRED.
      N=N-1
      AK=2**N-1+0.5
C     AK ABOVE IS PK INPUT TO A/D.
      AK=AK/ARPK
C     ARPK IS PK OUTPUT OF PREVIOUS D/A.
      AK=AK/(10.0**(S/20.0))
C     SCALE FACTOR FOR MAPPING NOW COMPLETE.
C     CHECK WHETHER MACHINE TRUNCATES OR ROUNDS.
      A=0.9
      IA=A
C     IA=0 IF TRUNCATION, IA=1 IF ROUNDING.
C     SIMULATE A/D CONVERTER:
      DO 1 I=1,MB
C     FIRST FIND POLARITY OF SAMPLE:
      ISN=1
      IF(AR(I).LT.0) ISN=-1
C     NOW ENCODE MAGNITUDE:
      AR(I)=AR(I)*ISN*AK
      IAR=AR(I)+0.5-(0.5*IA)
      IF(IAR.GT.(2**N-1)) IAR=2**N-1
1     AR(I)=IAR*ISN
      RETURN
      END
```

APPENDIX F THE COMBINED TRANSFORMATIONS USED FOR DIGITAL FILTER DESIGN

The approach adopted to program the automatic design of the digital filters has been outlined in Section 3.1.2.4. It was said there that each pair of conjugate poles in the analog low-pass prototype filter is transformed separately. The transform used combines the low-pass to band-pass transform and the bi-linear transform. The purpose of this appendix is to clarify the combined transform formulation, and hence its implementation in program *BUTTER*.

Stanley (1975, pp 195 et seq) has outlined the manipulations necessary to combine the low-pass to bandpass transform and the bi-linear transforms. Let quantities be defined as follows.

s	=	digital filter Laplace variable
p	=	analog low-pass prototype Laplace variable
$G(p)$	=	analog low-pass transfer function
$H(z)$	=	transfer function of digital band-pass filter
V_1, V_3	=	band edge frequencies of digital filter, normalised relative to the folding frequency.
V_2	=	normalised centre frequency of digital filter.
λ_r	=	analog low-pass reference radian frequency, imaginary component of p.
T	=	sampling period
z	=	e^{sT}

It is then algebraically convenient to define two derived quantities.

$$D = \lambda_r \cot \frac{\pi}{2}(V_3 - V_1) \qquad \text{(F-1)}$$

$$E = \frac{2 \cos (\pi/2)(V_3 + V_1)}{\cos (\pi/2)(V_3 - V_1)} \qquad \text{(F-2)}$$

The combined transform may then be a substitution procedure

$$H(z) = G(p)\Big|_{p = D\left[\frac{1 - Ez^{-1} + z^{-2}}{1 - z^{-2}}\right]} \qquad \text{(F-3)}$$

In program BUTTER it has been most convenient to reference the combined transform to the -3dB cut-off points of the analog low-pass prototype and the final digital band-pass filter. Also, it is convenient to use prototypes with cut-off angular frequency λ_r equal to unity. The prototypes are generated from the usual Butterworth pole pattern in the p-plane, with the equispaced poles now on a semi-circular locus of unit radius.

Each complex conjugate pole pair is transformed separately. This has been programmed on the following algebraic basis. Consider two poles with arbitrary location in the p-plane.

$$G_i(p) = \frac{1}{(p+a)\ (p+b)} \qquad \text{(F-4)}$$

where a and b are complex constants.

The transform substitution (F-3) may now be inserted, and the expression multiplied out. The algebra is lengthy, but gives

$$H_i(z) = \frac{1 - 2z^{-2} + z^{-4}}{C_1 + C_2 z^{-1} + C_3 z^{-2} + C_4 z^{-4} + C_5 z^{5}} \qquad \text{(F-5)}$$

where $C_1 = (D+a)\ (D+b)$

$C_2 = -DE\ (2D + a + b)$

$C_3 = 2D^2 - 2ab + D^2E^2$

$C_4 = -DE\ (2D - a - b)$

$C_5 = (D - a)\ (D - b)$ (F-6)

Now if the poles are at complex conjugate points, the coefficients C_i are purely real. Conveniently then

$$a + b = 2\ \mathrm{Re}\{a\} \qquad \text{(F-7)}$$

Also, because the poles lie on radius $\lambda_r = 1$,

$$ab = 1 \qquad \text{(F-8)}$$

The program calculates the coefficients using these simplifications of equations (F-6).

The gain of each four pole digital section is then normalised to unity at the centre frequency of the band-pass filter. The normalised centre frequency has been calculated from an expression given by Stanley (1975, p 196).

$$\tan^2 \left(\frac{\pi}{2} \cdot V_2\right) = \tan \left(\frac{\pi}{2} \cdot V_1\right) \cdot \tan \left(\frac{\pi}{2} \cdot V_3\right) \qquad \text{(F-9)}$$

The overall cascaded section band-pass filter thus also has unity gain at this centre frequency. This is useful if the filter is to operate repeatedly on a data array.

APPENDIX G NUMERICAL DATA AND RESULTS FROM SIMULATIONS

This appendix has no wide significance, but merely records some numerical details of representative simulations.

G.1 Default Values for Parameters Not Explicitly Defined

Random source set to produce 1024 variates, XMAX = 4,0. Thirteen points specified on the target distribution which is Gaussian.

Output X	Probability p(X)
0,00	0,4050
0,25	0,3900
0,50	0,3500
0,75	0,3050
1,00	0,2500
1,25	0,1820
1,50	0,1260
1,75	0,0870
2,00	0,0550
2,25	0,0310
2,50	0,0165
2,75	0,0093
3,00	0,0047

Sequence length = 32000 after skipping ISKIP = 10.

All filters 20 pole pairs. Sampling frequency 8 kHz.

Linear data converter resolutions 30 bits.

Send and signal filters 450-550 Hz.

Noise measurement filter 860-3400 Hz.

Digital attenuator setting X(dB) = 20dB.

Stimulus level dB reference = PCM peak input amplitude

All SQNR values and differences in dB.

A typical DATA file is listed below, together with suitable OS ...o control language (JCL)

Send side analog attenuator setting denoted S(dB). Receive side analog attenuator setting denoted SR (dB).

Negative attenuator settings indicate analog gain.

Attenuator setting X=20 , S= - SR.

SQNR figures used are corrected for noise measurement bandwidth, i.e. the SNRC value.

G.2 Probability Distribution Modelling

The random source target distribution has been labelled as RQPROB. The distribution actually achieved by the pseudo-random noise source has been labelled GENPROB, and this is the PDF of the signal at the input of the send filter. The following are typical plots of PDF's found in simulations. The intention is to emphasise the shape of the FLTPROB curve -- as this is the PDF of the filtered stimulus. Numerical details of the distributions are not emphasised here, and each graph was plotted from 64 data points only.

In all cases the FLTPROB is similar in shape, and appears approximately Gaussian. The distributions shown in Figures G1 to G5 were obtained using a 40 - pole send filter with a passband of 450 - 550 Hz. The filtered signal PDF is apparently not a strong function of the filter bandwidth. Figure G6 demonstrates an example where the send filter bandpass limits are 1 Hz to 3900 Hz; (the sampling frequency has been held at 8000 Hz).

When the send filter is fed from sinusoidal sources, the apparent FLTPROB is critically dependent upon the sampling frequency sub-harmonic effect. This is intuitively reasonable, and is not demonstrated here.

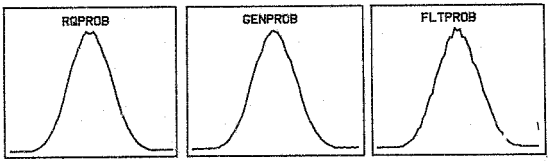


FIGURE G1 PDF'S FOR A GAUSSIAN RQPROB.

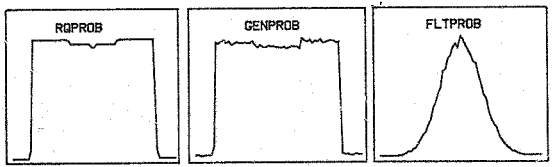


FIGURE G2 PDF'S FOR A RECTANGULAR RQPROB.

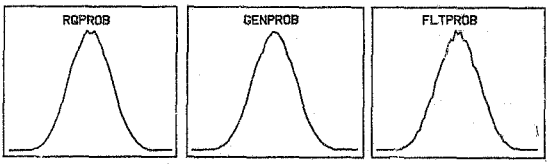


FIGURE G1 PDF'S FOR A GAUSSIAN RQPROB.

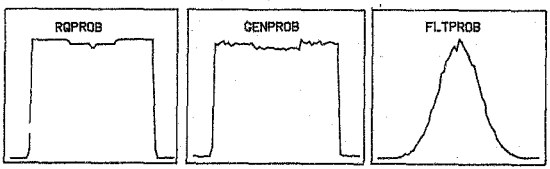


FIGURE G2 PDF'S FOR A RECTANGULAR RQPROB.

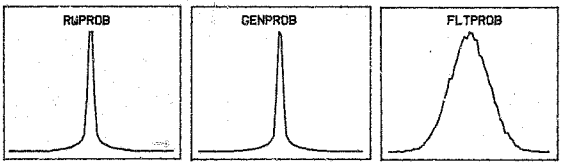


FIGURE G3 PDF'S FOR RQPROB AN ARBITRARY SPIKE.

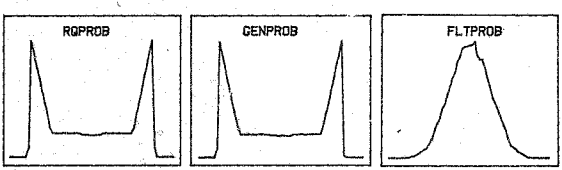


FIGURE G4 PDF'S FOR RQPROB SIMILAR TO THE DISTRIBUTION OF A SINUSOID.

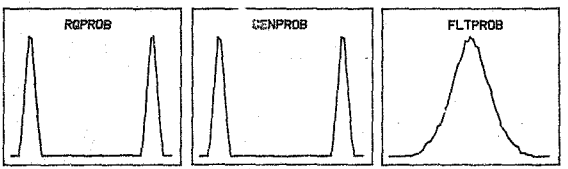


FIGURE G5 PDF'S FOR RQPROB A TWIN TRIANGULAR DISTRIBUTION.

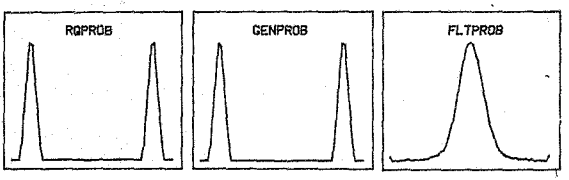


FIGURE G8 PDF'S FOR RQPROB A TWIN TRIANGULAR DISTRIBUTION, AND SEND FILTER BANDWIDTH 1 - 3000 HZ.

G.3 Ideal COMDAC SQNR versus Stimulus Level Characteristic

Stimulus Level	SQNR
0.0	11.120
-1.0	12.290
-2.0	13.667
-3	15.301
-4	17.173
-5	19.390
-6	22.060
-7	25.323
-8	28.777
-9	31.424
-10	33.919
-11	35.286
-12	35.298
-13	35.432
-14	35.339
-15	35.360
-20	35.253
-23	34.990
-25	35.054
-27	34.928
-30	34.595
-33	34.314
-35	33.948
-37	33.572
-40	32.902
-43	31.899
-45	31.261
-47	30.329
-50	28.626
-53	26.906
-55	25.638
-57	23.923
-60	20.974
-63	17.946
-65	15.940
-67	14.037
-70	11.059

G.4 Measured SQNR Variation with Test Length

S = - 0,135 dB (unintentional) SR = 0.

Stimulus Length (thousand samples)	SQNR (dB) stimulus -7dB	SQNR (dB) stimulus -13dB	SQNR (dB) stimulus -43dB
2	33,712	35,443	32,352
3	32,606	35,424	31,998
4	30,864	35,784	31,772
5	32,344	35,726	31,792
6	28,999	35,302	31,746
7	25,896	35,397	31,981
8	25,694	35,443	31,995
9	26,453	35,433	32,008
10	25,411	35,408	31,984
11	26,027	35,533	31,961
12	25,817	35,344	31,985
13	26,336	35,437	31,940
14	25,453	35,311	31,874
15	24,673	-	31,905
16	23,809	35,361	31,968
17	24,698	-	-
18	24,307	35,348	31,874
19	24,698	-	-
20	24,570	35,356	31,877
21	25,152	-	-
22	24,706	35,372	31,909
23	25,173	-	-
24	24,785	35,371	31,882
28	24,822	35,369	31,946
32	24,865	35,418	31,966

G.5 Measured SQNR as a Function of Stimulus Frequency Domain Parameters

G.5.1 Measured SQNR Variation with Stimulus Centre Frequency

Stimulus Band limits (Hz)	SQNR (dB) stimulus -7dB	SQNR (dB) stimulus -13dB	SQNR (dB) stimulus -43dB
1-101	36,203	35,389	31,803
100-200	33,775	35,300	31,924
200-300	30,311	35,406	31,945
300-400	22,078	35,236	31,982
400-500	23,698	35,224	32,005
450-550	25,323	35,432	31,899
500-600	22,734	35,342	31,890
600-700	21,863	34,986	31,866
700-800	18,545	20,179	20,022

G.5.2 Measured SQNR Variation with Stimulus Bandwidth

Filters initialised by 8000 samples.
Stimulus centre frequency 500 Hz.

Bandwidth (Hz)	SQNR (dB) stimulus -7dB	SQNR (dB) stimulus -13dB	SQNR (dB) stimulus -43dB
10	22,249	35,338	32,287
15	25,645	35,445	31,896
20	29,881	35,536	31,963
25	24,899	35,402	31,957
30	24,073	35,317	32,048
35	23,016	35,326	31,998
40	22,547	35,295	31,981
45	22,743	35,240	31,986
50	25,254	35,198	31,715
60	25,474	35,423	31,958
70	25,531	35,287	32,031
80	26,114	35,376	31,873
100	25,323	35,432	31,899
120	23,845	35,273	31,936
140	23,458	35,327	31,952
160	23,291	35,335	31,960
180	23,562	35,320	31,801

G.6 Data Converter Signal Level to Optimise Test Set SNR

Digital Attenuator X(dB)	SNR (dB)
2,78	11,29
5,48	19,81
6,27	22,64
7,13	26,21
8,05	30,28
9,02	34,22
10,01	40,31
11,01	54,01
11,21	58,26
11,41	62,43
11,61	62,53
11,81	62,28
12,01	62,17
13,01	61,34
14,00	60,45
14,99	59,53
15,99	58,62
16,99	57,70
18,00	56,75
19,00	55,75
20,00	54,83
25,00	49,90

SR = -S (dB) assumes unity gain PCM system.

G.7 Tests with Constrained Data Converter Resolution and Filter Order

G.7.1 Constrained Data Converter Resolution

	8-Bit A/D			10 - Bit A/D		
D/A Resolution	10	12	14	10	12	14
SQNR at -7dB	25,16	25,17	25,16	25,24	25,25	25,24
SQNR at -13dB	33,90	33,92	33,94	35,27	35,29	35,30
SQNR at -43dB	31,13	31,14	31,16	31,82	31,84	31,85

	12-Bit A/D			14-Bit A/D		
D/A Resolution	12	14	16	12	14	16
SQNR at -7dB	25,31	25,31	25,31	25,31	25,31	25,31
SQNR at -13dB	35,40	35,40	35,41	35,40	35,41	35,4
SQNR at -43dB	31,89	31,90	31,90	31,88	31,90	31,90

G.7 Tests with Constrained Data Converter Resolution and *Filter Order*

G.7.1 Constrained Data Converter Resolution

	8-Bit A/D			10 - Bit A/D		
D/A Resolution	10	12	14	10	12	14
SQNR at -7dB	25,16	25,17	25,16	25,24	25,25	25,24
SQNR at -13dB	33,90	33,92	33,94	35,27	35,29	35,30
SQNR at -43dB	31,13	31,14	31,16	31,82	31,84	31,85

	12-Bit A/D			14-Bit A/D		
D/A Resolution	12	14	16	12	14	16
SQNR at -7dB	25,31	25,31	25,31	25,31	25,31	25,31
SQNR at -13dB	35,40	35,40	35,41	35,40	35,41	35,42
SQNR at -43dB	31,89	31,90	31,90	31,88	31,90	31,90

G.7.2 Constrained Data Converter Resolution and Filter Order

	16/14 - Bit Set			
Pole Pairs per Receive Filter	8	10	12	16
SQNR at -7dB	25,02	25,25	25,28	25,30
SQNR at -13dB	33,48	35,16	35,38	35,41
SQNR at -43dB	30,90	31,77	31,87	31,90

	14/12 - Bit Set			
Pole Pairs per Receive Filter	10	12	14	16
SQNR at -7dB	25,26	25,29	25,30	25,30
SQNR at -13dB	35,15	35,36	35,40	35,40
SQNR at -43dB	31,77	31,87	31,89	31,89

	12/12 - Bit Set			
Pole Pairs per Receive Filter	10	12	14	16
SQNR at -7dB	25,26	25,29	25,30	25,31
SQNR at -13dB	35,14	35,36	35,38	35,39
SQNR at -43dB	31,76	31,86	31,88	31,88

APPENDIX H SQNR CURVE TRANSFORMATION PROGRAM: XFMG2S

```
C     PROGRAM TO XFM SQNR CURVES;
C     FROM SQNR PTS ESTABLISHED WITH GAUSSIAN STIMULI;
C     TO SQNR CURVE FOR GAMMA DISTRIBUTED STIMULI.
      REAL VG(100),QG(100),VOFFST(10)
      WRITE(6,200)
200   FORMAT(/,' ENTER NUMBER OF PTS MEASURED ON GAUSSIAN
SQNR:')
      READ(9,201)NPTSG
201   FORMAT(Y)
      WRITE(6,202)NPTSG
202   FORMAT(' FOR NEXT',I3,' LINES: ENTER STIMULUS LEVEL
& SQNR (DB):')
C     PTS SHOULD BE ENTERED IN ORDER OF DESCENDING LEVEL.
      DO 1 I=1,NPTSG
1     READ(9,203)VG(I),QG(I)
203   FORMAT(2Y)
      WRITE(6,204)
204   FORMAT(/,' ENTER NUMBER OF GAUSSIAN COMPTS GIVING GA
MMA APPROX:')
      READ(9,201)NAPPRX
      CALL TBLE(NAPPRX,SPREAD,VOFFST)
      DO 2 I=1,NPTSG
      VLOW=VG(I)+SPREAD
C     SPREAD SHOULD BE NEGATIVE.
C     SPREAD=VOFFST(NAPPRX-1) IN DB.
      IREM=I-1
      IF(VLOW.LE.VG(NPTSG)) GO TO 3
      CALL TRNSPT(NAPPRX,VG,QG,VSP,QS,DIFF,IREM,VOFFST,NPT
SG)
2     WRITE(7,205)VSP,DIFF,QS
205   FORMAT(3X,F9.5,',',F9.5,',',F9.5)
3     WRITE(6,206)IREM
206   FORMAT(/,T3,I3,' PTS ON XFMED CURVE WRITTEN TO OUTPU
T WORKSPACE.')
      STOP
      END
      SUBROUTINE TBLE(NAPPRX,SPREAD,VOFFST)
      REAL VOFFST(10)
C     ROUTINE TO SET UP LEVEL OFFSETS BETWEEN PTS;
C     SET OF OFFSET PTS REQUIRED TO GIVE EACH XFMED PT.
C     NON-EQUISPACED GAUSSIAN STIMULI LOOK-UP TABLE.
      IF(NAPPRX.NE.2) GO TO 2
      VOFFST(1)=20.0*ALOG10(0.31521/2.4935)
      GO TO 99
2     IF(NAPPRX.NE.3) GO TO 3
      VOFFST(1)=20.0*ALOG10(0.22978/0.897292)
      VOFFST(2)=20.0*ALOG10(0.22978/3.85505)
      GO TO 99
3     IF(NAPPRX.NE.4) GO TO 4
      VOFFST(1)=20.0*ALOG10(0.189133/0.667661)
      VOFFST(2)=20.0*ALOG10(0.189133/1.15860)
      VOFFST(3)=20.0*ALOG10(0.189133/11.00  )
      GO TO 99
```

```
4       IF(NAPPRX.NE.5) GO TO 5
        VOFFST(1)=20.0*ALOG10(0.167115/0.504753)
        VOFFST(2)=20.0*ALOG10(0.167115/0.862019)
        VOFFST(3)=20.0*ALOG10(0.167115/1.60016)
        VOFFST(4)=20.0*ALOG10(0.167115/28.7267)
        GO TO 99
5       IF(NAPPRX.NE.6) GO TO 6
        VOFFST(1)=20.0*ALOG10(0.153346/0.408269)
        VOFFST(2)=20.0*ALOG10(0.153346/0.695078)
        VOFFST(3)=20.0*ALOG10(0.153346/1.09217)
        VOFFST(4)=20.0*ALOG10(0.153346/2.06926)
        VOFFST(5)=20.0*ALOG10(0.153346/143.38)
        GO TO 99
6       IF(NAPPRX.NE.7) GO TO 7
        VOFFST(1)=20.0*ALOG10(0.144249/0.342228)
        VOFFST(2)=20.0*ALOG10(0.144249/0.577662)
        VOFFST(3)=20.0*ALOG10(0.144249/1.04371)
        VOFFST(4)=20.0*ALOG10(0.144249/1.04402)
        VOFFST(5)=20.0*ALOG10(0.144249/3.24052)
        VOFFST(6)=20.0*ALOG10(0.144249/203.49)
        GO TO 99
7        IF(NAPPRX.NE.8) GO TO 8
        VOFFST(1)=20.0*ALOG10(0.137901/0.299004)
        VOFFST(2)=20.0*ALOG10(0.137901/0.485968)
        VOFFST(3)=20.0*ALOG10(0.137901/0.805636)
        VOFFST(4)=20.0*ALOG10(0.137901/1.16469)
        VOFFST(5)=20.0*ALOG10(0.137901/1.18230)
        VOFFST(6)=20.0*ALOG10(0.137901/5.41517)
        VOFFST(7)=20.0*ALOG10(0.137901/91.8962)
        GO TO 99
8        IF(NAPPRX.NE.9) STOP
        VOFFST(1)=20.0*ALOG10(0.134458/0.254911)
        VOFFST(2)=20.0*ALOG10(0.134458/0.521966)
        VOFFST(3)=20.0*ALOG10(0.134458/0.529311)
        VOFFST(4)=20.0*ALOG10(0.134458/1.04331)
        VOFFST(5)=20.0*ALOG10(0.134458/1.26076)
        VOFFST(6)=20.0*ALOG10(0.134458/1.35517)
        VOFFST(7)=20.0*ALOG10(0.134458/9.97449)
        VOFFST(8)=20.0*ALOG10(0.134458/46.4940)
99      NOFFS=NAPPRX-1
        SPREAD=VOFFST(NOFFS)
        RETURN
        END
        SUBROUTINE TRNSPT(NAPPRX, VG,GG,VSP,GS,DIFF,IREM,VOF
FST,NPTSG)
        REAL VG(100),GG(100),VOFFST(10),VGINT(10),GGINT(10),
NSESP
C       IPNTR POINTS TO HIGHEST LEVEL STIMULUS PT INVOLVED I
N XFM.
        IPNTR=IREM+1
        CALL INTERP(NAPPRX,VG,GG,VGINT,GGINT,VOFFST,IPNTR,NP
TSG)
C       DB REF TO UNIT VOLTAGE.
C       INTERPOLATED VGINT NO LONGER IN DB.
        NSESP=0.0
        VSP=0.0
```

```
      DO 1 I=1,NAPPRX
      ANSE=VGINT(I)*(10.0**(-QGINT(I)/20.0))
      NSESP=NSESP+ANSE**2
1     VSP=VSP+VGINT(I)**2
      NSESP=NSESP/NAPPRX
      VSP=VSP/NAPPRX
      AVSP=VSP/NSESP
      QS=10.0*ALOG10(AVSP)
      VSP=10.0*ALOG10(VSP)
C     FIND SQNR GAUSSIAN AT LEVEL EQUIV TO SPEECH PT:
      CALL SQNRG(VG,QG,VSP,QGV,NPTSG)
      DIFF=QGV-QS
      RETURN
      END
      SUBROUTINE INTERP(NAPPRX,VG,QG,VGINT,QGINT,VOFFST,IP
NTR,NPTSG)
      REAL VG(100),QG(100),VGINT(10),QGINT(10),VOFFST(10)
C     ROUTINE TO SET UP ONE SET OF PTS TO GIVE ONE XFMED P
T.
C     DEFINES THE VGINT(PURE NUMBER) & QGINT(DB) TO BE USE
D FOR NEXT PT.
      VGINT(1)=VG(IPNTR)
      NOFFS=NAPPRX-1
      DO 1 I=1,NOFFS
      II=I+1
1     VGINT(II)=VGINT(1)+VOFFST(I)
C     VOFFST(I) NEGATIVE VALUES IN DB.
      DO 2 I=1,NAPPRX
      VSPEC=VGINT(I)
      CALL SQNRG(VG,QG,VSPEC,QGVSPC,NPTSG)
      QGINT(I)=QGVSPC
2     VGINT(I)=10.0**(VGINT(I)/20.0)
      RETURN
      END
      SUBROUTINE SQNRG(VG,QG,VSPEC,QGVSPC,NPTSG)
      REAL VG(100),QG(100)
C     GIVEN PTS ON GAUSSIAN SQNR CURVE;
C     THIS ROUTINE PERFORMS A LINEAR INTERPOLATION;
C     RETURNS SQNR CORRESPONDING TO SPECIFIED STIMULUS LEV
EL.
C     SPECIFIED AND RETURNED VALUES ALL IN DB.
C     DB REF TO UNIT VOLTAGE.
      IF(VSPEC.GT.VG(1)) STOP
C     ASSUME VG(I) DECREASE WITH DECREASING I.
      NL=NPTSG-1
      DO 1 I=1,NL
      IPNTR=I+1
      IF(VSPEC.GT.VG(IPNTR)) GO TO 2
1     CONTINUE
      STOP
2     IPNTR2=IPNTR-1
      SLOPE=(QG(IPNTR2)-QG(IPNTR))/(VG(IPNTR2)-VG(IPNTR))
      DELX=VSPEC-VG(IPNTR)
      QGVSPC=QG(IPNTR)+SLOPE*DELX
      RETURN
      END
```

APPENDIX I A SIMULATION PROGRAM FOR DETERMINISTIC STIMULI: SINES

I.1 A Program to Set Up the Simulation: INITSN

```
C     SET UP DATA FILE FOR SINES:
      WRITE(6,508)
508   FORMAT(/,'  BUILD UP MODULATING ENVELOPE:',/,'  ENTE
R NUMBER OF SINUSOIDAL COMPONENTS:')
      IPLO=1
      READ(9,100)NG
200   FORMAT(2Y)
100   FORMAT(Y)
      IDUM=0
300   FORMAT(3Y)
      WRITE(7,112)NG,IDUM
      WRITE(6,500)NG
500   FORMAT('/,' ENTER ENVELOPE LEVELS ON NEXT',I3,' LINES
:')
      DO 1 I=1,NG
      READ(9,100)E
1     WRITE(7,731)E
      WRITE(6,736)
736   FORMAT(/,' FILTER MODULATING ENVELOPE; ENTER NO. PO
LES IN L-P PROTO:')
      READ(9,100)NLPS
      WRITE(6,111)
111   FORMAT(/,' ENTER NUMBER OF SINUSOIDAL SUB-CARRIERS
REQUIRED:')
      READ(9,100)ISTP
      WRITE(7,112)NLPS,ISTP
      WRITE(6,113)
113   FORMAT(/,' ENTER SAMPLING FREQ., LOWER, UPPER MODUL
ATING FREQ. :')
      READ(9,735)FS,F1,F3
735   FORMAT(3Y)
      WRITE(7,710)FS,F1,F3
112   FORMAT(1X,I5,3X,I5)
      WRITE(6,737)
737   FORMAT(/,' ENTER LOWEST, HIGHEST SUB-CARRIER FREQS.
 :')
      READ(9,200)F1,F3
      WRITE(7,710)FS,F1,F3
      WRITE(6,725)
725   FORMAT(/,'  NOW SET-UP THE SEND FILTER: ',/)
      WRITE(6,101)
101   FORMAT(/,' ENTER LOWER, UPPER BAND EDGE FREQS IN H
Z. : ',/)
      READ(9,200)F1,F3
4     WRITE(6,102)
102   FORMAT(/,' ENTER READ ORDER (EVEN) OF LOW-PASS PRO
TOTYPE. : ',/,' 0=NO FILTER REQUIRED:'))
      READ(9,100)NLPS
      WRITE(6,712)
```

```
112   FORMAT(/,'  SIMULATION TO STOP WHEN SEND PROSS. FOUN
D? 1=YES: ',/)
      READ(9,100)ISTP
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
710   FORMAT(1X,F10.3,2(2X,F10.3))
      WRITE(6,726)
726   FORMAT(/,'  ENTER NUMBER OF BITS RESOLUTION OF D/A :
 ',/)
      READ(9,100)IBDA
      IF(IPL0.EQ.1) IBDA=6
      WRITE(6,727)
727   FORMAT(/,'  TEST SIGNAL LEVEL TO BE X DB DOWN ON PEA
K INPUT.',/,'  ENTER REQUIRED X (DB): ',/)
      READ(9,100)RX
      WRITE(7,728)IBDA,RX
728   FORMAT(T4,I4,3X,F10.3)
      IF(ISTP.EQ.1) STOP
      WRITE(6,730)
730   FORMAT(/,'  ENTER ANALOG ATTENUATOR SETTING TO BE SI
MULATED (DB):',/)
      READ(9,100)S
      WRITE(7,731)S
731   FORMAT(T5,F10.3)
      WRITE(6 732)
732   FORMAT(/,'  NOW SET-UP SIGNAL FILTER ON RECEIVE SIDE
.',/,)
      WRITE(6,101)
      READ(9,200)F1,F3
      WRITE(6,102)
      READ(9,100)NLPS
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
      WRITE(6,734)
734   FORMAT(/,'  NOW SET-UP RECEIVE NOISE FILTER.',/)
      WRITE(6,101)
      READ(9,200)F1,F3
      WRITE(6,102)
      READ(9,200)NLPS
      WRITE(7,112)NLPS,ISTP
      WRITE(7,710)FS,F1,F3
      STOP
      END
/DATA
/INC DAT.SN
```

I.2 Simulation Program for Deterministic Stimuli: SINES

```
      DIMENSION E(512),AR(32767)
C     STIMULUS LENGTH FIXED.
      MB=32767
C     DEFINE MODULATION ENVELOPE:
      READ(5,112)NG,IDUM
112   FORMAT(1X,I5,3X,I5)
C     NG=NUMBER OF GAUSSIAN COMPONENTS.
      DO 1 I=1,NG
1     READ(5,730)E(I)
      IA=(32768.0/2.0)/NG
C     SET UP ENVELOPE WAVEFORM:
C     SET UP MIDDLE FOR OVERLAP:
      DO 5 I=16000,17000
5     AR(I)=E(NG)
C     SET UP FIRST HALF OF ENVELOPE:
      DO 2 I=1,NG
      DO 3 J=1,IA
      IPNTR=J+(I-1)*IA
3     AR(IPNTR)=E(I)
2     CONTINUE
C     SET UP SECOND HALF OF ENVELOPE:
      IB=IA*NG
      DO 4 I=1,IB
      IPNTR=32767-I
      IPNTR=IPNTR+1
4     AR(IPNTR)=AR(I)
C     STEPPED ENVELOPE WAVEFORM NOW DEFINED.
      WRITE(6,708)
708   FORMAT('  STEPPED ENVELOPE WAVEFORM DEFINED.')
C     CALCULATE ENVELOPE MEAN BEFORE FILTERING:
      SUM=0.0
      DO 6 I=1,MB
6     SUM=SUM+AR(I)
      SUM=SUM/MB
C     NOW FILTER ENVELOPE WAVEFORM:
      READ(5,112)NLPS,NSUB
C     NSUB=NUMBER OF SUB-CARRIERS.
      READ(5,710)FS,F1,F3
      ISW=1
      IF(NLPS.NE.0)CALL FLTS(AR,MB,NLPS,FS,F1,F3,ISW,FPWR)
      IF(NLPS.NE.0) CALL MEAN(AR,MB,SUM)
      IF(NLPS.NE.0) WRITE(6,707)
707   FORMAT('  ENVELOPE FILTERED.')
C     GENERATE SINUSOIDAL SUB-CARRIERS: THEN MODULATE:
      READ(5,710)FS,F1,F3
C     FS SAME VALUE AS BEFORE;
C     F1 AND F3 SUB-CARRIERS AT EXTREMES OF BAND.
      CALL GEN(AR,FS,F1,F3,NSUB)
      WRITE(6,709)
709   FORMAT('  STIMULUS GENERATED.')
```

```
C     NOW FILTER IF REQUIRED:
      READ(5,112)NLPS,ISTP
      READ(5,710)FS,F1,F3
710   FORMAT(1X,F10.3,2(2X,F10.3))
      ISW=1
      FPWR=0.0
      IF(NLPS.NE.0)CALL FLTS(AR,MB,NLPS,FS,F1,F3,ISW,FPWR)
       IF(NLPS.NE.0) WRITE(6,711)
711   FORMAT(/,'  SEQUENCE FILTERED BY SEND FLTS. ',/)
      READ(5,728)IBDA,RX
      CALL QUANTO(AR,MB,IBDA,RX)
       WRITE(6,729)
728   FORMAT(I4,I4,3X,F10.3)
729   FORMAT(/,'  QUANTO COMPL       ',/)
      IDUM=1
      WRITE(7,727)IDUM
727   FORMAT(I3,I3)
      CALL PRBPLT(AR,MB)
      IF(ISTP.EQ.1) STOP
      READ(5,730)S
      CALL IDLPCM(AR,MB,IBDA,S)
       WRITE(6,731)
730   FORMAT(T5,F10.3)
731   FORMAT(/,'  IDLPCM COMPLETED.',/)
      CALL CONVRT(AR,MB)
       WRITE(6,732)
732   FORMAT(/,'  CONVERT COMPLETED.',/)
      READ(5,112)NLPR,ISTP
      READ(5,710)FS,F1,F3
      F1S=F1
      F3S=F3
      ISW=0
      CALL FLTS(AR,MB,NLPR,FS,F1,F3,ISW,FPWR)
      SPWR=FPWR
       WRITE(6,733)
733   FORMAT(/,'  REC. SIGNAL FILTER COMPLETED.',/)
      READ(5,112)NLPR,ISTP
      READ(5,710)FS,F1,F3
      CALL FLTS(AR,MB,NLPR,FS,F1,F3,ISW,FPWR)
      QPWR=FPWR
      SNR=20.0*ALOG10(SPWR/QPWR)
      QPWR=QPWR*(FS/2.0)/(F3-F1)
      SPWR=SPWR-QPWR*(F3S-F1S)/(FS/2.0)
      SNRC=20.0*ALOG10(SPWR/QPWR)
       WRITE(6,740)SNR,SNRC
740   FORMAT(/,'  SNR= ',F8.3,T25,' SNRC= ',F8.3,' (DB).',
/)
        STOP
        END
```

```
      SUBROUTINE FLTS(AR,M,NLPS,FS,F1,F3,ISW,FPWR)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE BUTTER(N,C,FS,F1,F3)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE QUANTO(AR,M,IBDA,X)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE IDLPCM(AR,M,IBDA,S)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE CONVRT(AR,M)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE PRBPLT(AR,MB)
C     THIS ROUTINE AS IN "PSPEC".
      SUBROUTINE GEN(AR,FS,F1,F3,K)
      DIMENSION AR(32767)
      T=1.0D0/FS
      DELF=(F3-F1)/(K-1)
      PI=4.0D0*DATAN(1.0D0)
      DO 1 I=1,32767
      DACC=0.0D0
      DO 2 J=1,K
      JJ=J-1
      DARG=(2.0D0*PI*(F1+JJ*DELF)*(I-1)*T)
      DT=SIN(DARG)
      DACC=DACC+DT
2     CONTINUE
      AR(I)=AR(I)*DACC
1     CONTINUE
C     SUM OF SINUSOIDS EVALUATED AT A PT., THEN MOD'N.ATED.
      RETURN
      END
      SUBROUTINE MEAN(AR,MB,SUM)
      DIMENSION AR(32767)
      DO 1 I=1,MB
1     AR(I)=AR(I)+SUM
      RETURN
      END
```

Author Goedhals Anton Beau
Name of thesis A Basis For Digital Instrumentation To Measure Quantisation Distortion In Speech Pulse Code Modulation Systems. 1979

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.