THE DESIGN OF A CASCADED DIGITAL/ANALOG FILTER FOR DIGITAL TRANSMISSION SYSTEMS

Uswatte Liyanage Anton Eric Marian
Priyakumar Perera

A Thesis

in

The Faculty

of

Engineering and Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Engineering at
Concordia University
Montreal, Quebec
Canada

April 1982

ABSTRACT

THE DESIGN OF A CASCADED DIGITAL/ANALOG FILTER FOR DIGITAL TRANSMISSION SYSTEMS

by

Uswatte Liyanage Anton Eric Marian Priyakumar Perera

Equalization procedures for a class of widely known conventional filters that have potential applications at the transmit side of digital transmission systems are presented in this thesis. A binary (digital) transversal filter is cascaded in front of an analog filter so as to predistort the input digital signal in order to remove intersymbol-interference caused by the analog filter. The latter is used to shape the power spectrum of the digital signal to be transmitted. The design of the transversal filters is based on the 'a priori' knowledge of the analog filter characteristics. Three methods of computing tap-gain coefficients of the digital device are presented. A computer simulation program is described, which may be used to observe the improvement of the eye diagram at the analog filter output when the transversal filter is cascaded in front of it. This allows for choosing among off-the-shelf conventional analog filters for use in digital transmission systems without actually building any hardware units. By eliminating intersymbol-interference in the transmitted signal the digi-

tal device improves the bit error rate performance of the system in an additive white gaussian noise environment by as much as 4 dB, simultaneously increasing the in-band to out-of-band energy ratio of the filtered signal, as witnessed by the test results.

The noise used in the performance tests may be random or pseudo-random in its nature. Computer simulation and hardware test results of a simple and low cost pseudo-random generator-based noise source that exhibits the gaussian amplitude characteristic are presented in the first part of the thesis. It is shown experimentally that the bit error rate performance of a baseband digital transmission system in additive gaussian noise environment generated by random and pseudo-random noise sources shows little difference in their effects on the system.

ACKNOWLEDGMENTS

I thank Dr. K. Feher and Dr. M.N.S. Swamy for their keen interest in this study. Their guidance and encouragement have been of great value. I thank Dr. Feher also for suggesting this research topic and insisting not only on the theoretical but also on the practical aspects of the project.

My thanks are due also to Drs. J. Huang and R. El-Attar for their help in computer simulation; to Messrs M. Gendron, M. El-Torky, A. Brind'Amour (U. of Ottawa) and Dr. S. Natarajan for their assistance in hardware. I am grateful to Concordia University Computer Centre for providing the word processing facilities to document this manuscript.

The patience and understanding of my wife, Diana, during the preparation of this thesis are greatly appreciated. I thank her for the encouragement she gave me throughout my study and for editing this manuscript.

Finally, I dedicate this work to two generations in my family: my son, Dinuka, and my mother and father.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AF	analog filter
AWGN	additive white gaussian noise
BER	bit error rate
BTF	binary transversal filter
b/s	bits per second (prefixes k for kilo and M for mega)
CMOS	complementary metal-oxide semiconductor
CRO	cathode ray oscilloscope
DC	direct current
dB	decibel(s) [$10 \log(P_2/P_1)$ for power ratio and $20 \log(V_2/V_1)$ for voltage ratio]
dB/div	decibel(s) per division
E_b/N_0	energy per bit to noise density ratio
erf(x)	error function of x, = $(2/\sqrt{\pi})\,(\int_0^x e^{-t^2} dt)$
erfc(x)	complementary erf(x), = $(2/\sqrt{\pi})\,(\int_x^\infty e^{-t^2} dt)$
f	frequency variable in Hz
f_b	bit rate or data rate
f_0 or f_c	3 dB cut-off frequency in Hz
H:	horizontal scale (in photographs)
H(s, f or ω)	network transfer function in s, f or ω
Hz	Hertz - units of frequency (prefixes k for kilo and M for mega)
h(t)	impulse response (of a network)
IC	integrated circuit
ISI	intersymbol-interference
i	an integer
k	an integer

kHz/div	kiloHertz per division
LPF	low pass filter
m	an integer
ms/div	millisecond(s) per division
N	noise power (or an integer in a summation)
NRZ	non-return to zero
N_0	single-sided noise power density
n	an integer
OA	operational amplifier
P(e)	probability of error
PRBS	pseudo-random binary sequence(s)
p.d.f.	probability density function
RC	resistor-capacitor (combination)
rad/sec	radians per second - units of angular frequency (prefix k for kilo)
rms	root-mean-square
S	signal power
SRL	shift-register length
S/N	signal-to-noise ratio
s	Laplace variable
sec	second(s)
T	element or bit interval of digital signal
TF	transversal filter
V:	vertical scale (in photographs)
V/div	volt(s) per division
κ	ratio of f_b:f_c
σ	standard deviation
ω	angular frequency variable in rad/sec

ω_0	normalized cut-off frequency (angular)
Σ_n and $\Sigma_{i=1}^{k}$	summation over all n and over i = 1, 2, ..., k
Σ'_n	summation over all n with deletion of n = 0 term
μs/div	microsecond(s) per division

Chapter 1

INTRODUCTION

Digital communications systems have advanced rapidly in the past two decades. This rapid growth can be attributed to a great extent to the development of digital circuitry. With the advancement of digital computers, the size of hardware has been reduced immensely together with their costs, at the same time increasing efficiency. The availability of high speed devices has also contributed to the increase of system capacity in digital communications systems, which was never achievable with the analog counterpart. However, during the actual transmission the discrete character of digital signals vanishes due to the filtering process of unwanted high frequencies. Filtering is an essential process in a communication system so as to utilize the available frequency spectrum prudently. Thus, digital signals become continuous waveforms during their passage through the transmission channel.

Conventional analog devices such as filters often find application in digital transmission systems. Much work has been done over the years and is still being done to improve these types of devices with active components replacing the earlier passive ones. In this way, their implementation has become cheaper and more efficient with an enormous reduction in size. A major drawback, however, in active analog filters is their group delay characteristics often undesirable for digital signals. Both amplitude and group delay characteristics, as such, cause intersymbol-interference (ISI) on digital signals. In order to alleviate such distortion an equalization process is essential. Numerous papers have been written on a variety of complex equalization schemes to equalize channels, filters, etc. New and more effective methods are still being explored and proposed in the literature. This thesis presents the work done on a low cost and simple equalization method that can be used on a conventional analog filter intended to be used at the transmit end of a digital transmission system.

Before a communication system is put into service, a system model is critically evaluated based on known or expected engineering parameters of that system. In a laboratory environment a noise source is used to simulate the thermal noise encountered in such systems. It is well known that amplitudes of random front-end receiver noise present in communication systems have a normal (gaussian) distribution function. Manufacturers offer, often expensive, gaussian noise sources with prescribed characteristics suitable for such simulations. In evaluating a laboratory model of a baseband communication system a simple and low-cost noise source, built in the laboratory, could provide equally well the service rendered by a costly unit. Hence, the first part of the thesis describes the implementation of a pseudo-random generator-based gaussian noise source presented in Chapter 2. It contains a literature survey with mathematical background, computer simulation and hardware implementation of a noise source that exhibits gaussian amplitude characteristics. The test results of this noise source are compared with that of a conventional noise source.

Chapter 3 is a study on transversal filters. The transversal filter is a non-recursive type of a digital filter. It has found application in a wide area of digital systems. A literature survey containing a mathematical analysis of the filter for application in communication systems is presented in this chapter. The design of such a filter cascaded with a conventional active analog filter, which can be used at the transmit end of a transmission system, is described in Chapter 4. The transversal filter is used here as an equalizer for the analog one that is used to shape the spectrum of the signal. It shows how a readily available (off-the-shelf) analog filter could be used in a digital communication system without causing distortions in the signal, merely by cascading a simple transversal filter so that the signal is predistorted before passing it through the analog filter, which, in turn, shapes the signal spectrum. Methods of finding the optimum values of filter parameters are presented. Computer simulation results of a baseband system (only the transmit end) with and without a predistorter cascaded to the analog filter are compared with the results of the hardware model tested in the laboratory. The entire model is evaluated with regard to the bit error rate performance in an additive white gaussian noise environment. The evaluation is extended to using the laboratory-built gaussian noise source.

The summary and conclusion of the thesis is given in Chapter 5. Suggestions for future research on the cascaded filter design based on the work here are presented. A list of references and appendices terminate the presentation. The appendices contain the description and computer simulation programs, additional mathematics and further procedures of hardware implementation.

Chapter 2

GAUSSIAN NOISE GENERATED BY PSEUDO-RANDOM BINARY SEQUENCES

2.1 Introduction

It is well known that excitation of systems by noise is a valuable aid in dealing with problems in system identification and performance evaluation. Physical devices such as resistors and diodes are sometimes used as sources of random noise. However, such devices suffer from statistical variability between finite samples of their noise output. Often it is adequate, or even advantageous, to employ deterministic periodic signals with the essential statistical properties of random noise.

The deterministic character of the pseudo-random binary sequences (PRBS), which are also called pseudo-noise sequences, or m-sequences, or chain codes [Lipson et al., 1976], has proved itself very useful in generating random-like noise with desired probablity density functions. The characteristic qualities of a desirable noise are,

i. the bandwidth of the noise power spectrum should enclose that of the system under test, thus ensuring that all modes of the system are excited;

ii. the noise should be pseudo-random and repeatable so that meaningful tests may be reproduced under varying conditions for, then, the noise source will appear ergodic when its time average is taken over a specified interval;

iii. the statistics of the noise should be known precisely, thereby removing a source of uncertainty in experimental results;

iv. the noise signal should have negligible serial autocorrelation after a short definable interval.

All of the above properties are possessed by maximum-length linear binary sequences generated by shift-register circuits with modulo-2 addition and feedback except for the reservation that the autocorrelation function is periodic [Golomb, 1964; Roberts & Davis, 1966; Cumming, 1967]. The PRBS obtained by shift-register technique, therefore, provide the feasibility of generating random noise that has an amplitude distribution approximating a gaussian distribution. This is done by bandlimiting (lowpass filtering) the PRBS.

This chapter presents a brief review of the theory of PRBS generation, the simulation of these sequences on a general purpose digital computer, and the results of hardware implementation based on those of computer simulation. Finally, it is explained how such sequences can be used for the performance test of transversal filters described in the next chapter.

2.2 The Pseudo-Random Binary Sequence

The pseudo-random binary sequence (PRBS) differ from the truly random binary sequence (TRBS) as follows:

i. a TRBS is non-periodic whereas a PRBS repeats itself after some suitably long sequence;

ii. in many random binary processes, e.g., a random telegraph wave, the transition from the logic 1 state to the logic 0 state, and vice-versa, can occur at any time, and the state at any instant of time is independent of that at any other instant. In the pseudo-random process the binary level transitions can occur only at specific clock pulse times, separated by intervals during which the binary state is fixed. (Here, the state during the fixed time interval is independent of that during neighbouring intervals.)

A periodic binary sequence will be classified as a PRBS if it satisfies the following conditions [Hampton,1965]:

i. in each period the number of logic 1s (ones) must not exceed that of logic 0s (zeros) by more than one, or vice-versa;

ii. in each period there must be twice as many sequences of 1s or 0s of length n as those of length (n+1);

iii. the autocorrelation function must be of the form as shown in Fig.2.1(b), i.e., peaked in the middle ($\tau = 0$) and tapering off rapidly at both ends. Such a binary sequence with a sufficiently long period can be used essentially like true random noise. Periodic binary

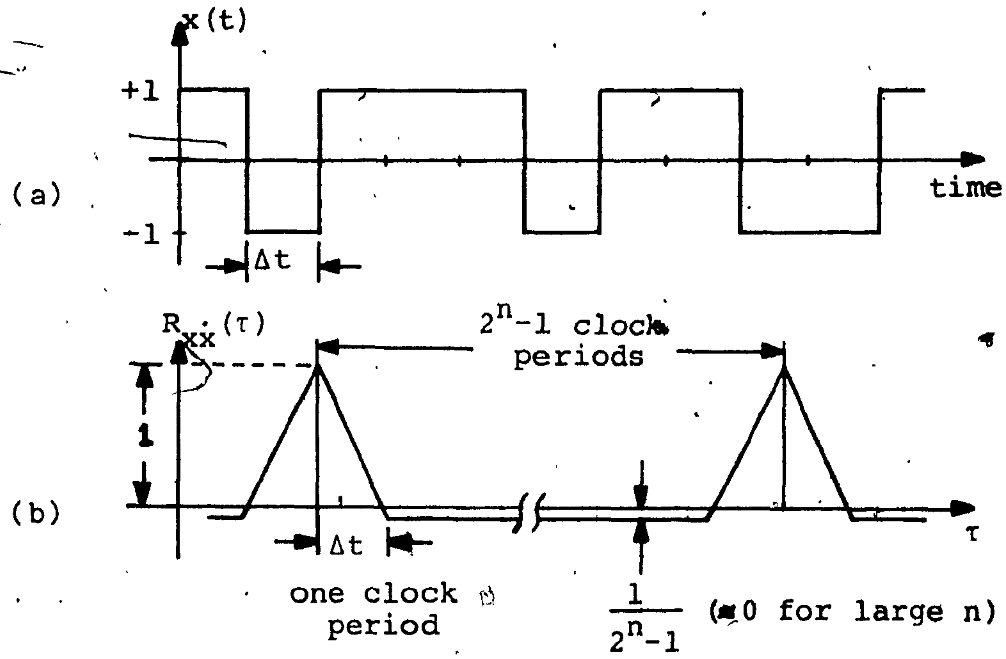


Fig.2.1(a) Waveform and (b) Time-Autocorrelation Function of a Maximum Length Sequence.

sequences may easily be obtained from a digital shift-register with modulo-2 adder feedback as explained in the next section.

2.2.1 The Feedback Shift-Register

The feedback shift-register is a binary (digital) circuit consisting of a shift-register and one or more modulo-2 adders whose final mod-2 sum is fed back to the first stage

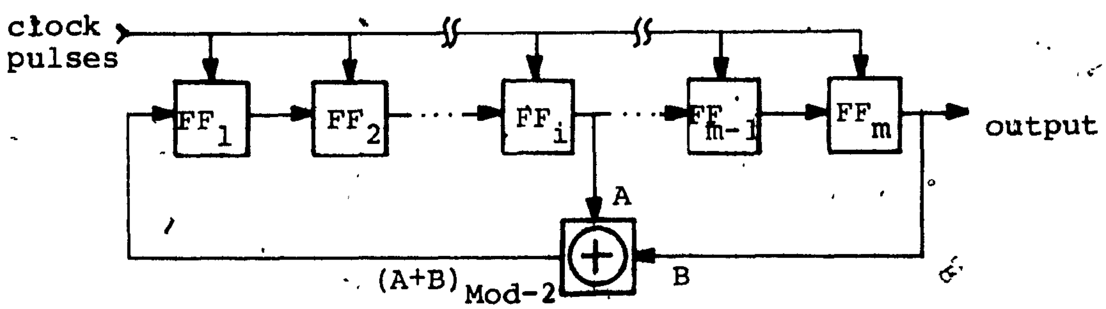


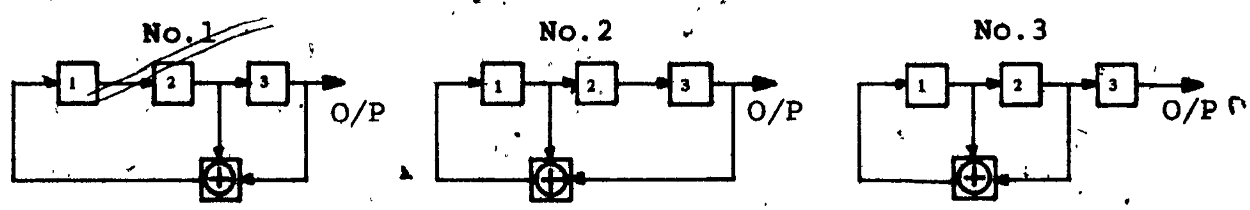
Fig.2.2 Feedback Shift-Register with Modulo-2 Adder.

of the shift-register as shown in Fig.2.2. Modulo-2 addition generates the sum $(A+B)_{Mod-2}$ of any two binary inputs A and B according to the following rule:

A	B	$(A+B)_{Mod-2}$
0	0	0
0	1	1
1	0	1
1	1	0

As can be seen from the above rule, mod-2 addition can be implemented logically with an exclusive-OR circuit. The shift-register consists of cascaded flip-flops driven at the desired rate by external clock pulses. (The clock pulses are not shown in all the figures, hereafter.)

Fig.2.3 illustrates three 3-stage shift-register periodic sequence generators and their corresponding sequences. It can be seen that the generators marked No.1 and No.2 are



CLOCK PULSE NUMBER	No. 1 STAGE 1 2 3		No. 2 STAGE 1 2 3		No. 3 STAGE 1 2 3
Init. Cond.	1 0 1		1 0 1	one	1 0 1
1	1 1 0		0 1 0	period	1 1 0
2	1 1 1		0 0 1		0 1 1
3	0 1 1	one	1 0 0	repeat	1 0 1
4	0 0 1	period	1 1 0	periods	. . .
5	1 0 0		1 1 1		. . .
6	0 1 0		0 1 1		. . .
7	1 0 1	repeat	1 0 1		. . .
.	. . .	periods	. . .		. . .
.	. . .		. . .		. . .
Init. Cond.	0 0 0		0 0 0		0 0 0
1	0 0 0		0 0 0		0 0 0
2	0 0 0		0 0 0		0 0 0
.	. . .		. . .		. . .
.	. . .		. . .		. . .
n	0 0 0		0 0 0		0 0 0

Fig.2.3 Three-Stage Periodic Binary Sequence Generators When Initial Condition is 101 or 000.

periodic every $2^3-1 = 7$ bits, whereas the generator marked No.3 is periodic every 3 bits. In each case the periodic series is completely determined by the initial state of the flip-flops and feedback connections. The resulting sequences for an all-zero initial condition are also shown. The generators No.1 and No.2 are known as "maximal-length sequence (or m-sequence) generators" for m = 3.

It is easy to show that the maximum length of any sequence produced by a shift-register is (2^m-1), where m is the number of flip-flop stages in the shift-register. For an m-stage generator there are 2^m possible states. The all-zero state can be ruled out as an inadmissible condition, because with mod-2 addition, each succeeding state would also be all zero as shown in Fig.2.3. Therefore, the sequence is periodic with a maximum of (2^m-1) bits. Some important properties of these maximum length sequences are outlined in Section 2.2.2. A method, supported by mathematics, of obtaining the m-sequences from a shift-register is described in Appendix A.

2.2.2 Properties of Pseudo-Random Sequences

This section outlines some very important properties of pseudo-random sequences which are valuable in their application in the context of this chapter.

Let $h(x)$ be a fixed primitive* polynomial of degree m, and let S_m be the set consisting of the PRBS obtained from $h(x)$, together with the sequence of (2^m-1) zeros, i.e., 00...0, denoted by $\underline{0}$. These PRBS are the (2^m-1) different segments,

$$a_i a_{i+1} \ldots a_{i+2^m-2}, \qquad i = 0, 1, \ldots, 2^m-2$$

of length (2^m-1) from the output of the shift-register specified by $h(x)$.

<u>Property I - The Shift Property</u>: If $\underline{b} = b_0 b_1 \ldots b_{2^m-2}$ is any pseudo-random sequence in S_m, then any cyclic shift of $\underline{b}$, say $b_j b_{j+1} \ldots b_{2^m-2} b_0 \ldots b_{j-1}$, is also in S_m.

<u>Property II - The Recurrence</u>: Suppose $h(x) = \Sigma_{i=0}^{m} h_i x^i$, with $h_0=h_m=1$, $h_i=0$ or 1 for $0<i<m$. Any pseudo-random sequence $\underline{b} \in S_m$ satisfies the recurrence,

$$b_{i+m} = h_{m-1} b_{i+m-1} + h_{m-2} b_{i+m-2} + \ldots + h_1 b_{i+1} + b_i \qquad (2.1)$$

for i = 0, 1, ..., m. Conversely, any solution of (2.1) is in S_m. There are m linearly independent solutions to eqn.(2.1); hence, m linearly independent sequences in S_m.

----

* See Appendix A for the definition of primitive polynomial.

<u>Property III- The Window Property:</u> If a window of width m is slid along a pseudo-random sequence in S_m, each of the (2^m-1) non-zero binary m-tuples is seen exactly once (see Fig.2.4 for the case m = 4). This follows from the fact that h(x) is a primitive polynomial.

... 0 0 0 1 $\boxed{0\ 0\ 1\ 1}$ 0 1 0 1 1 1 1 ...

Fig.2.4 The Window Property: Every Non-Zero 4-Tuple is Seen Once.

The difficulty in seeing the property at the ends of this sequence could be overcome if the sequence is written in a circle as shown in Fig.2.5.

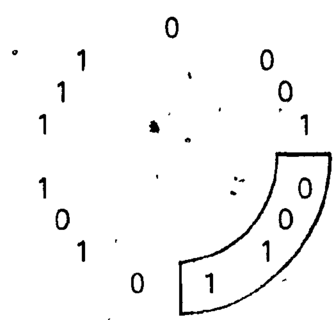


Fig.2.5 The Pseudo-Random Sequence in Fig.2.4 is Written in a Circle.

<u>Property IV - Half 0s and Half 1s:</u> Any pseudo-random sequence in S_m contains 2^{m-1} logic 1s and $(2^{m-1}-1)$ logic 0s.

Property V - The Addition: The sum of two sequences in S_m (formed component-wise, modulo-2, without carries) is another sequence in S_m.

Property VI - The Shift-and-Add: The sum of a pseudo-random sequence and a cyclic shift of itself is another pseudo-random sequence. (From properties I and V)

Property VII - The Autocorrelation Function: The autocorrelation function $R_{xx}(\tau)$ of a pseudo-random sequence is given by [Haykin ,Ch.2, 1978],

$$R_{xx}(\tau) = \begin{cases} 1 - |\tau| \dfrac{2^m}{2^m-1} & \text{for } |\tau| \leq 1 \text{ bit} \\ -\dfrac{1}{2^m-1} & \text{for } |\tau| \geq 1 \text{ bit} \end{cases}$$

Fig.2.1(b) shows this function.

Property VIII - Runs*: In any pseudo-random sequence, one-half of the runs have length 1, one-quarter have length 2, one-eighth have length 3, and so on, as long as these fractions give integral numbers of runs. In each case the number of runs of logic 0s is equal to that of logic 1s.

---

* A run, here, is defined to be a maximal string of consecutive identical symbols (either logic 0s or logic 1s).

Properties III, IV, VII and VIII justify the name pseudo-random sequences, for these are the properties that one would expect from a sequence obtained by tossing a fair coin (2^m-1) times. Such properties make pseudo-random sequences very useful in a number of applications such as range finding [Evans and Hagfois, 1968; Golomb, 1964; Pettengill and Skolnik, 1970], synchronization, modulation [Golomb, 1964], scrambling [Feistel et al., 1975; Henriksson, 1972; Nakamura and Iwadare, 1972; Savage, 1967], etc.

2.3 Generation of Gaussian Noise from a Pseudo-Random Binary Sequence Source

According to the central limit theorem of statistics, the distribution of the sum of a number of independent random variables tends toward gaussian as the number of terms in the sum is increased. McFadden [1959] showed that when a random telegraph wave is lowpass filtered, the filter output has a gaussian amplitude distribution if the filter time constant is much greater than the reciprocal of the mean count rate, i.e., average number of zero crossings per second. This phenomenon may be understood by noting that the areas between the zero crossings of the random telegraph wave are independent random variables. If there are (m+1) zero crossings within the memory span of the LPF, then its output approximates the sum of m independent random variables. Thus, as m increases the distribution approaches gaussian, as stated by the central limit theorem.

The areas between the zero crossings of pseudo-random sequence may also be considered to be independent random variables. Therefore, severe bandlimiting through a LPF should again produce an output with a normal (gaussian) amplitude distribution. However, an additional constraint in the pseudo-random case is that the period of the sequence be very long, relative to filter time constant. It must be noted that the random variable associated with the pseudo-random sequence is discrete, while that of the random telegraph wave is continuous.

2.3.1 On Summing the Pseudo-Random Binary Sequences

It was discussed earlier that lowpass filtering of the PRBS is equivalent to summation of the independent random variables (in this case, the areas between zero crossings). However, alternatives to lowpass filtering have been proposed in the literature [Kramer, 1965; Douce and Healy, 1969].

One method is to add together a number of delayed versions of the same signal, thus, simulating the action of the lowpass filter, as shown in Fig.2.6. An important cons-

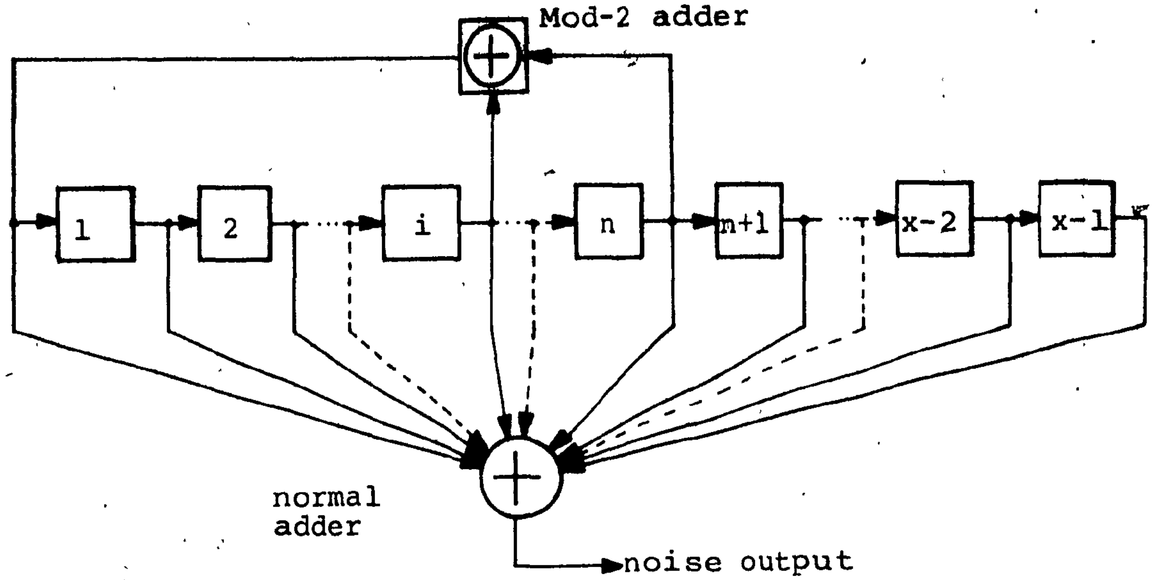


Fig.2.6 Delayed Version of PRBS Noise Generator.

traint in this method is that $x < 2n$, where x and n are integers as shown in Fig.2.6. The other method is to replace the analog lowpass filter by a digital allpass filter. By obtaining the Fourier transform of the autocorrelation function it has been shown that the half-power bandwidth of the PRBS noise generator is approximately eight times that obtained by lowpass filtering [Rowe and Kerr, 1970]. This is done by adding together m past bits of the PRBS resident in the shift-register weighted by the sequence $h_{AP}(k)$, the impulse response of the optimized allpass filter, as shown in Fig.2.7.

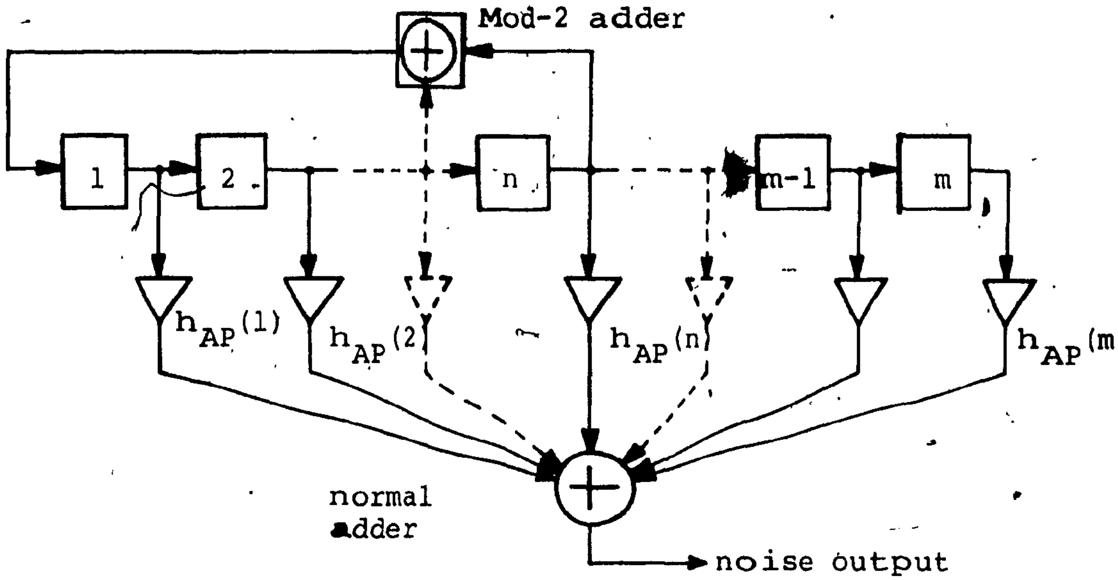


Fig.2.7. Weighted Sequence Model of PRBS Noise Generator.

2.3.2 Simulated Gaussian Noise

A program was run on a general purpose digital computer to obtain noise samples that result from low pass filtering a stream of maximal length PRBS. The process of generating and filtering the sequences is explained in Appendix C which describes the computer simulation. As described there, the long stream of PRBS was processed in sections, which were combined later to re-construct the continuous stream. Depending upon the length of the shift-register, the length of the stream was varied and a simple resistor-capacitor (RC) low pass filter was used to process the sequences. The cut-off frequency of the filter was also varied on each simulation run, keeping the bit rate fixed, to observe the variation in the results. The distribution of amplitudes of filtered sequences were plotted on each run to observe their density functions. The mean and standard deviation of these functions were computed and used to obtain the equivalent gaussian (normal) probability density function (p.d.f.). Thus, on each plot are histograms of both pseudo-random p.d.f. and gaussian p.d.f. superimposed on each other. These histograms are an aid for observing the behaviour of the distribution as filter parameter is varied and for comparing the pseudo-random noise density function with the gaussian p.d.f. These plots are shown in Figures 2.8 through 2.12. In these figures, the quantity $\kappa = f_b/f_c$, where f_b is the bit rate in kilobits per second (kb/s) and f_c is the filter cut-off frequency in kiloHertz (kHz), is used as a normalized measuring scale of comparison among the histograms. In all the histograms the hyphen (-) represents the histogram bars of filtered PRBS; the plus sign (+) represents the expected gaussian function for the same mean and variance; and upper scale letter I represents the overlap between the two histograms.

Figure 2.8 shows two sets of histograms, one of which is for $\kappa = 40$ and the other is for $\kappa = 32$. It is seen that the shift-register length (SRL) has an impact on the amplitude p.d.f. of the filtered PRBS. Although the filtered PRBS density function does not have the exact gaussian shape, an increase in SRL causes an improvement in the shape of the function to a large extent. This is because the period of the sequences is now longer (SRL = 9 vs. SRL = 7) providing "more randomness" in the sequences. Another aspect of the improvement is the skewness of the function. When SRL = 7 the function is skewed towards left of the gaussian peak and this is not the case when SRL = 9.

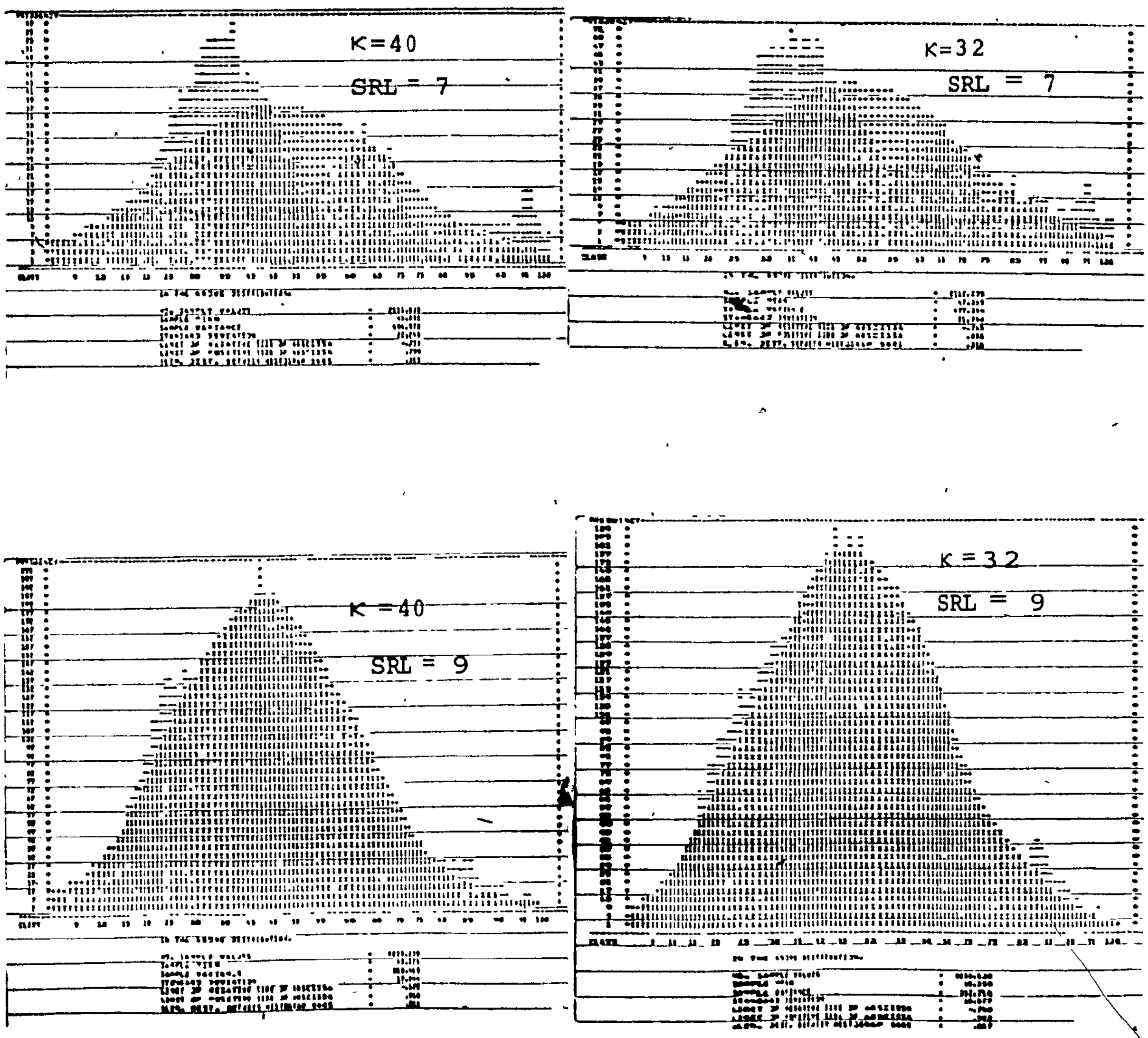


Fig.2.8 Histograms of Filtered PRBS When SRL and Filter Cut-Off Frequency are Varied; SR was Clocked at 32 kHz. (SRL = 7 & 9 and f_c = 0.8 & 1.0 kHz)

Figure 2.9 shows the cases for κ = 24.6 and κ = 22.9 with the same pair of SRLs. Here, also, similar observations can be made with regard to the shape of the density functions. The filtered PRBS density function almost follows the gaussian in the larger SRL case, except at its peak, for both values of κ. However, the shorter SRL does not provide good results even with these values of κ.

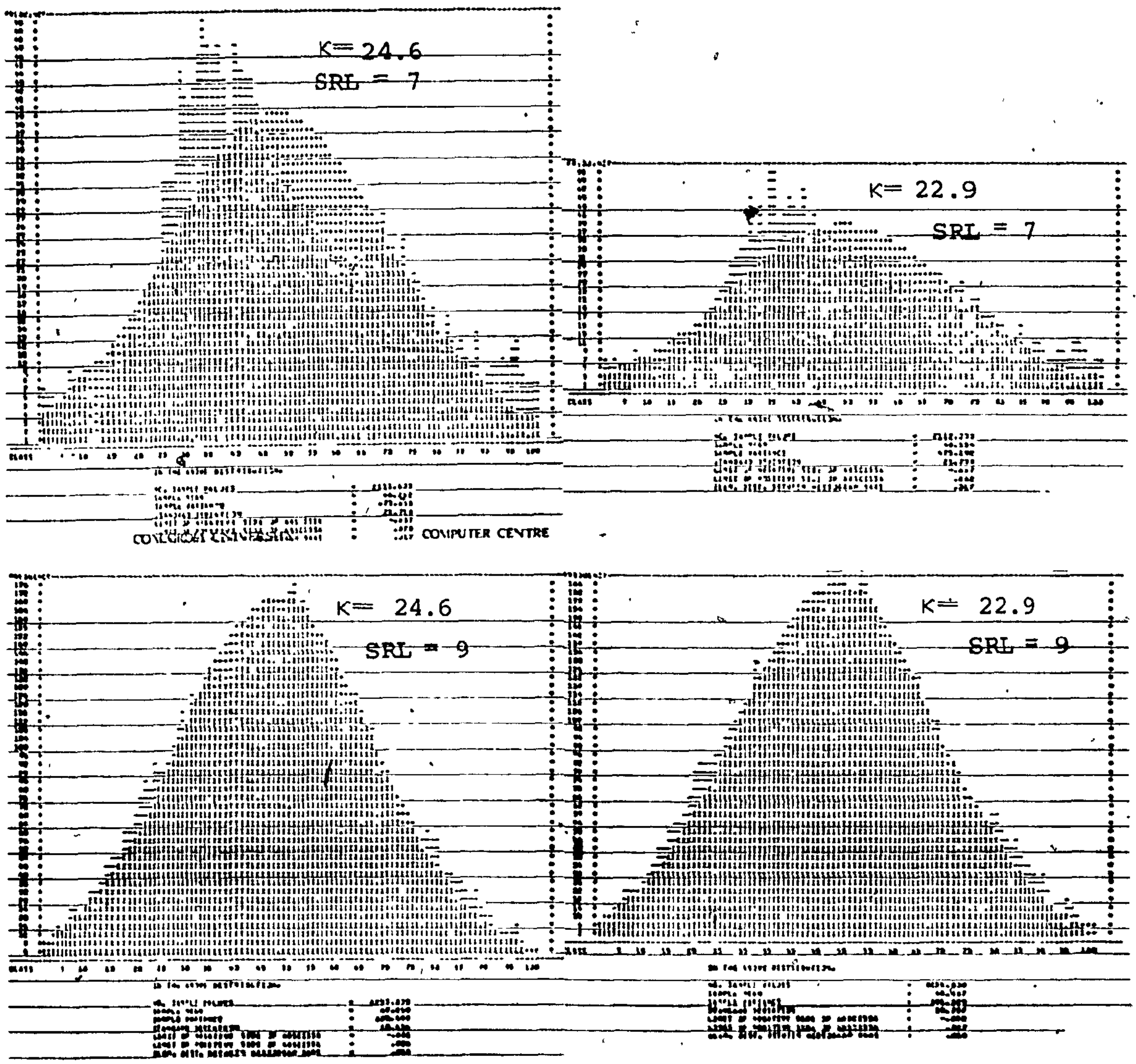


Fig.2.9 Histograms as in Fig.2.8 but f_c = 1.3 & 1.4 kHz.

A further improvement in the density functions is observed in Fig.2.10. With SRL increased to 10 and κ set to 22.9 a good approximation to the gaussian p.d.f. is obtained, as depicted in the upper left histogram. However, when κ is set to 20 very good approximations are observed in both cases of SRL equal to 9 and 10. An undesired shape still persists when SRL = 7 as shown by the upper right histogram.

If κ is decreased further (κ = 16.0, 10.7, 3.2 and 1.0) the PRBS p.d.f. drifts away completely from the gaussian characteristic, as shown by Figs.2.11 and 2.12, until the function flattens in the center leaving two spikes at the extremeties with almost 0.5 probability density. These spikes relate to the equal probability of occurrence of logic 1 and logic 0 in the PRBS, now that the filter is wide-band (κ = 1.0 means that a filter with f_c = 32 kHz processing a binary data stream of 32 kb/s). This result (two spikes) is in agreement with the Property IV of maximal length PRBS as discussed in Section 2.2.2.

Some statistical data extracted from these simulation results (histograms) are shown in Table 2.1. It is interesting to note that as κ is decreased from 40.0 to 1.0 (in 9 steps) the observed sample mean of each histogram increases

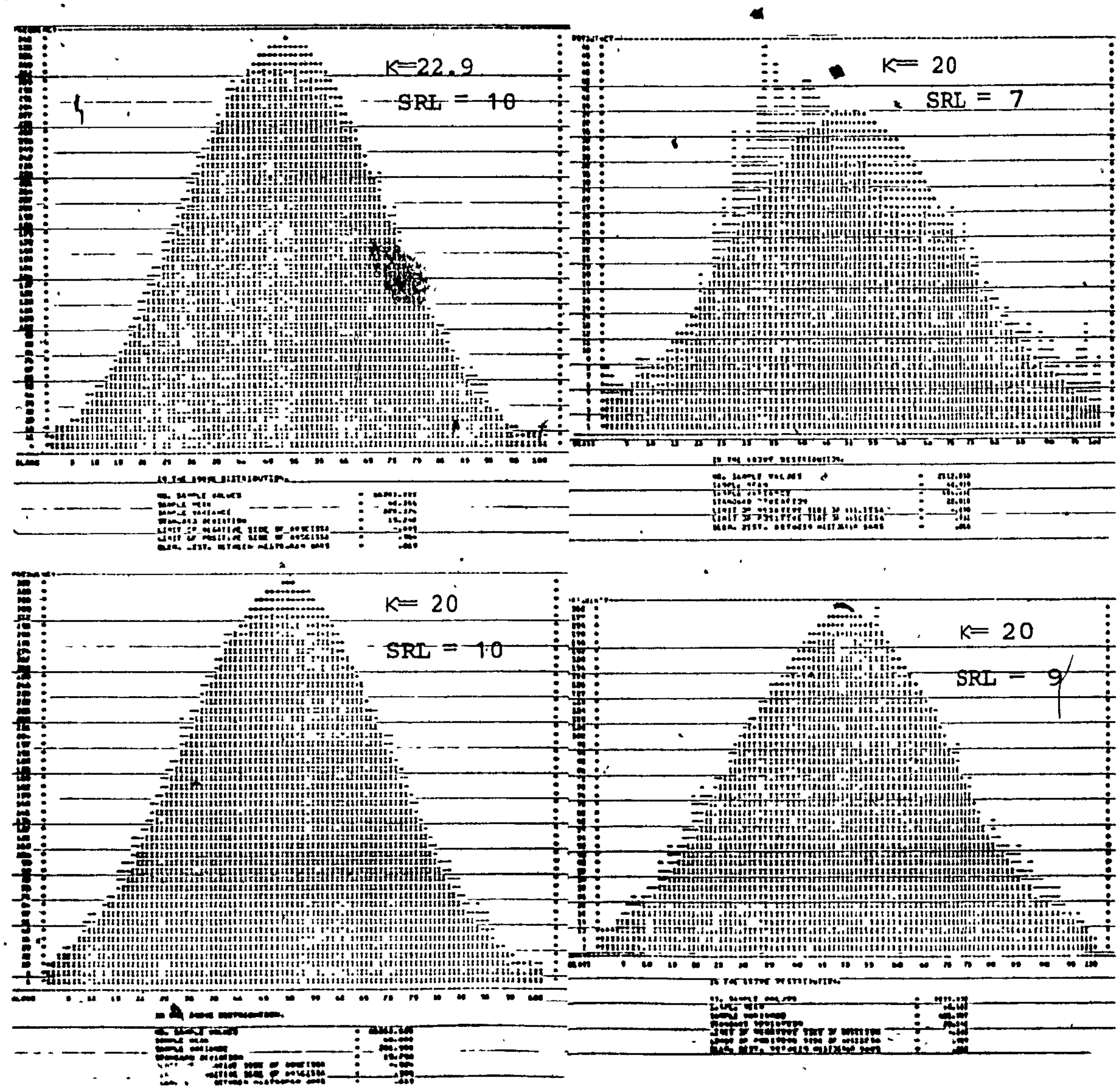


Fig.2.10 Histograms When SRL = 7, 9 & 10
and f_c = 1.4 & 1.6 kHz.

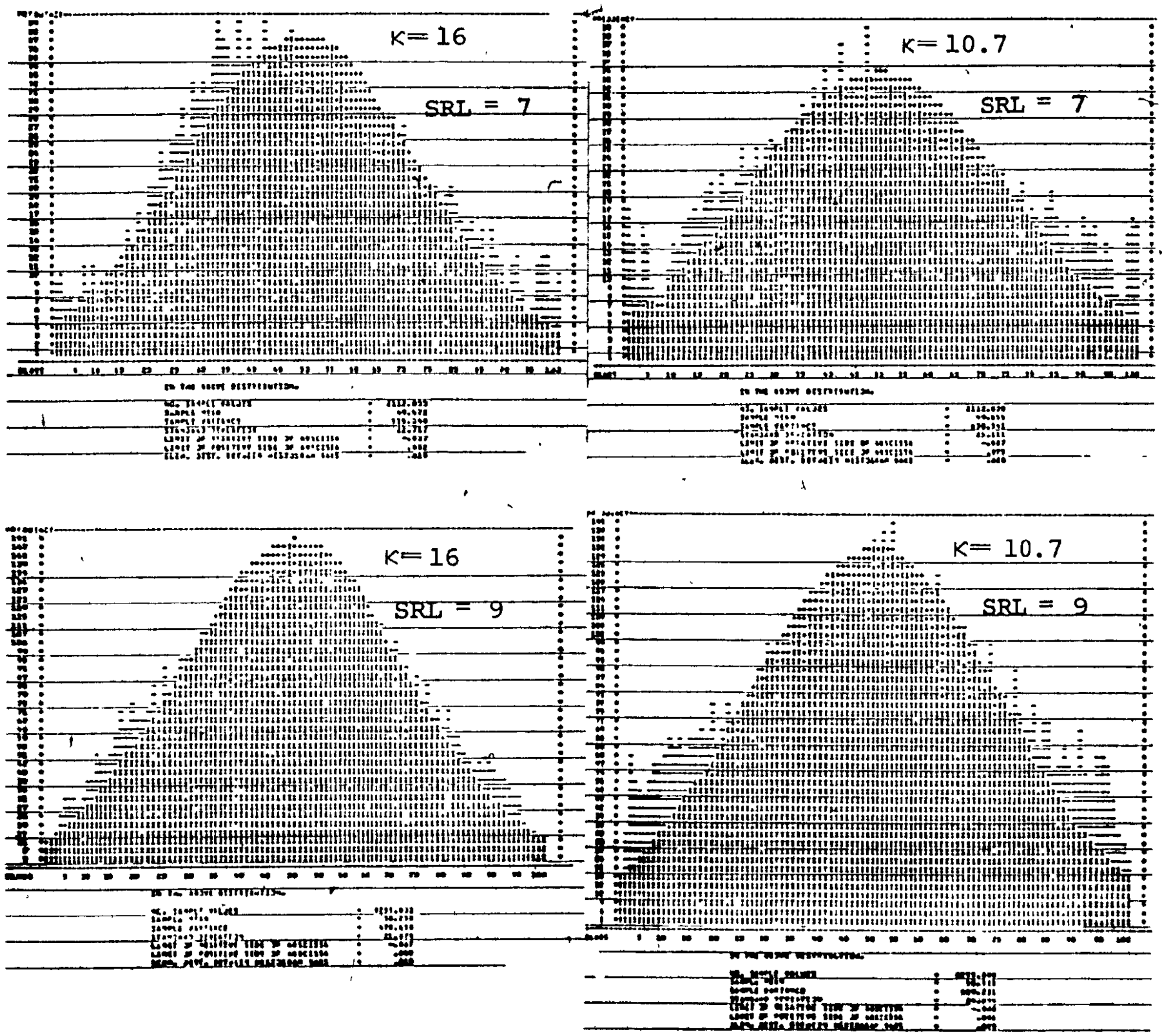


Fig.2.11 Histograms When SRL = 7 & 9
and f_c = 2.0 & 3.0 kHz.

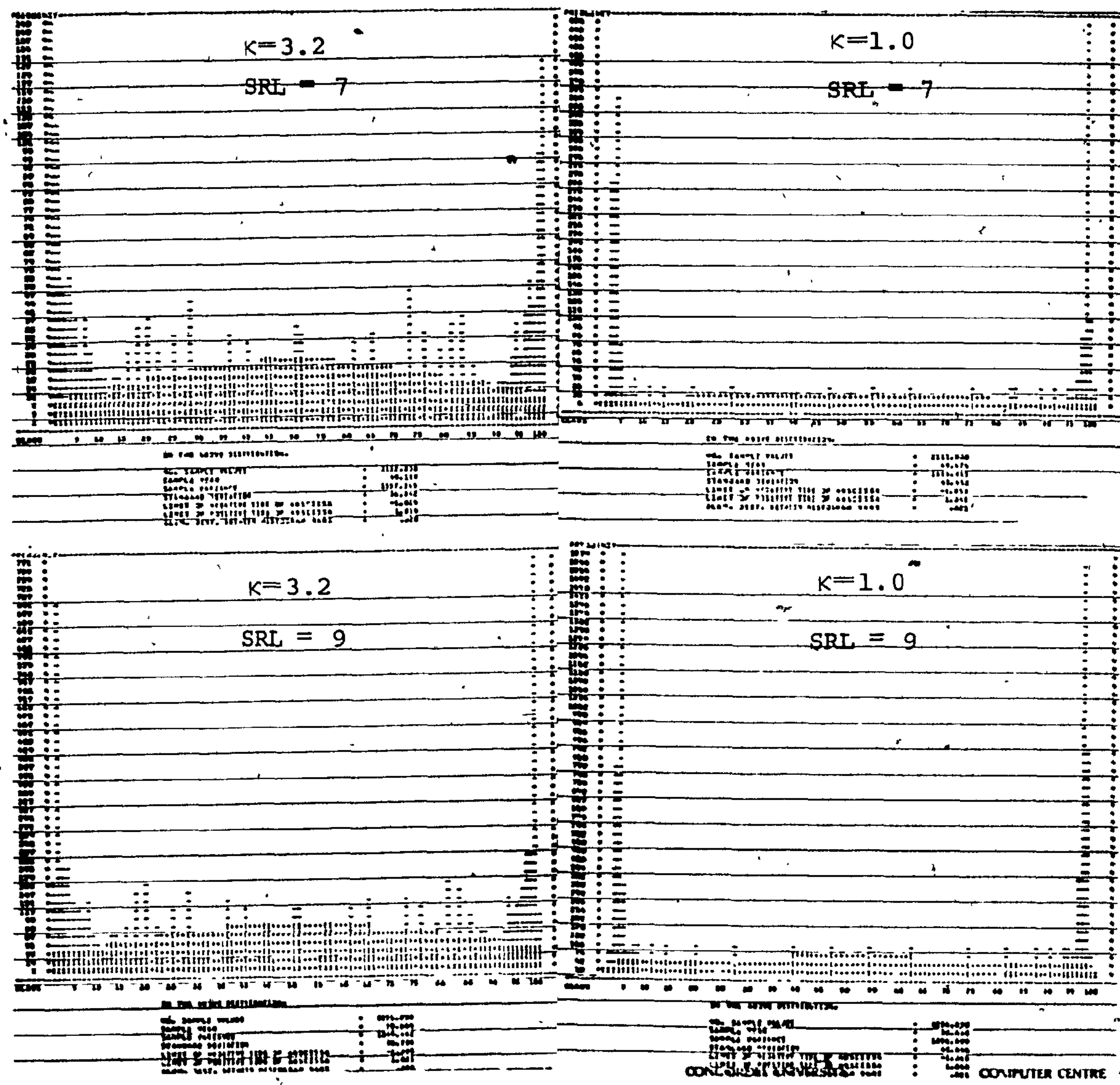


Fig.2.12 Histograms as in Fig.2.11 but f_c = 10.0 & 32.0 kHz.

approaching the expected* value.

----

* In the simulation the binary data are generated with the amplitude levels +1.0 and -1.0 of equal probability; the expected mean of amplitude p.d.f. is, therefore, 0. However, for the purpose of plotting the histograms the abscissa was scaled to have values from 1 to 100, represen-

Table 2.1 The Mean and Standard Deviation Observed in the Histograms

SHIFT-REGISTER LENGTH[1]			7	9	10	7	9	10
STEP	κ[2]	SAMPLE MEAN EXPECTED	SAMPLE MEAN OBSERVED			STANDARD DEV. OBSERVED		
1	40.0	50.0	45.8	43.0	-	22.2	17.8	-
2	32.0	50.0	47.1	45.4	-	21.8	18.4	-
3	24.6	50.0	48.2	47.9	-	21.7	19.4	-
4	22.9	50.0	48.6	48.5	48.4	21.8	19.8	19.2
5	20.0	50.0	49.0	49.5	48.8	22.0	20.5	19.8
6	16.0	50.0	49.5	50.3	-	22.7	21.9	-
7	10.7	50.0	49.7	50.5	-	25.1	24.9	-
8	3.2	50.0	49.2	50.7	-	36.8	36.8	-
9	1.0	50.0	49.5	50.7	-	43.5	43.1	-

Note: 1. The shift-register length was set to 10 only in two cases. The simulation program requires a prohibitively long time for the processing of signal sections at this SRL value and beyond.

2. κ is the ratio of bit rate, f_b kb/s, to cut-off frequency, f_c kHz, of the LPF used to filter the sequences. Here, f_b was fixed at 32 kb/s but f_c was varied.

However, the standard deviation decreases to some minimum value as κ is decreased, but increases as κ is further decreased until unity. The best aprroximation, therefore, takes place when $\kappa = 20$. It is difficult at this point to infer, from simulation results, the extent of the gaussian

----

ting the above levels, so that the expected mean now becomes 50.

nature of the amplitude p.d.f. in terms of its standard deviation, σ. In other words, up to how many σ does the p.d.f. behave gaussian? This question could have been answered if the histogram envelopes were smooth curves. The smoothness is achieved as more and more samples (random) are obtained. Because of the time-limit imposed by the computer system when processing a very long PRBS, SRL was set to a maximum of 10 stages. This produced an m-sequence of only 1023 bits in a period. The randomness of samples in this sequence is believed to be insufficient in this application. However, this question is addressed again when smoother p.d.f. curves are presented as hardware test results in the next section.

To sum up the simulation results, it is observed that as long as the shift-register is at least 10 stages long and the cut-off frequency of the low pass filter is about one-twentieth the bit rate, it is possible to generate pseudo-random noise that exhibits a gaussian amplitude p.d.f.

2.3.3 Hardware Implementation and Results

The computer simulation results reported in the previous section affirmed that random-like noise samples that have a gaussian amplitude p.d.f. can easily be generated by using a feedback shift-register and a low pass filter. Such a type of noise source is a cost effective alternative to often expensive, manufactured noise generators used by laboratories for performance evaluations of communication systems. This section describes the hardware implementation of a low cost pseudo-random noise generator that exhibits a gaussian amplitude distribution among its noise (output) samples. The test results are presented in the form of photographs taken on instruments displays.

A variety of feedback arrangements for generating maximal length PRBS have been reported [Bardou, 1981; Harvey, 1978], which pull away the circuit from locking into an "all-zero" state, especially at start-up. These arrangements range from using discrete gates or counters that detect an all-zero state at any instant to a simple RC network that introduces a delay into a gate at start-up, all of which introduce a logic 1 into the first stage of the shift-register to start the generation of PRBS. Figures 2.13 through 2.15 show three such configurations that yield a maximal sequence length for a given number of flip-flop stages in the shift-register, never allowing the circuit to stay locked in an all-zero state.

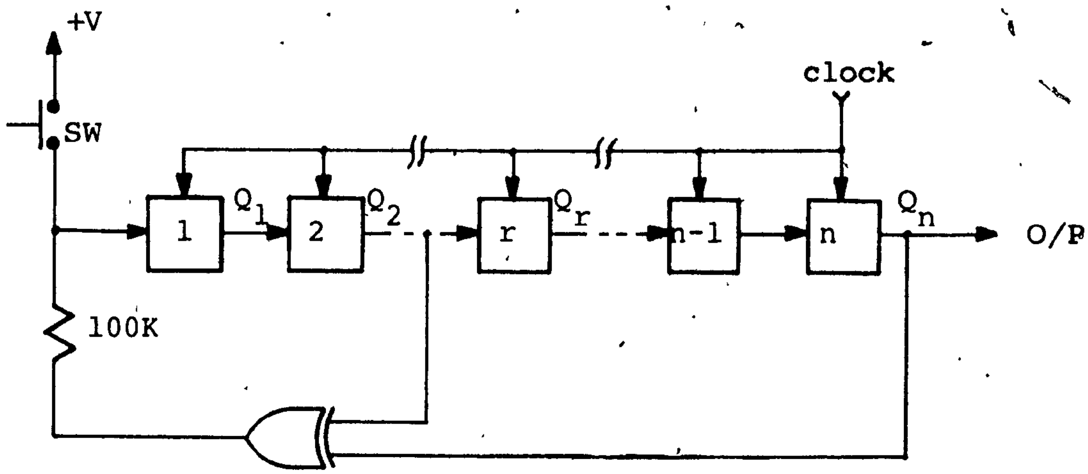


Fig.2.13 Configuration I: Manual Operation of Switch SW Forces a Logic 1 into the First Stage.

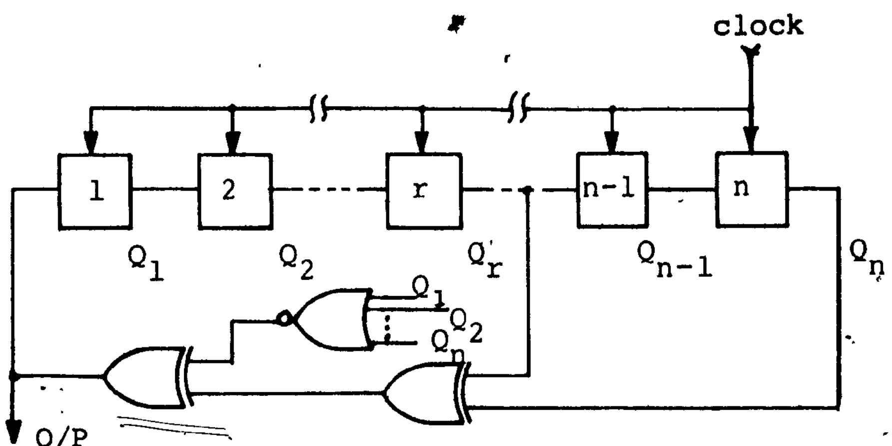


Fig.2.14 Configuration II: NOR Gate Detects All-Zero State and Inserts a Logic 1 into the First Stage.

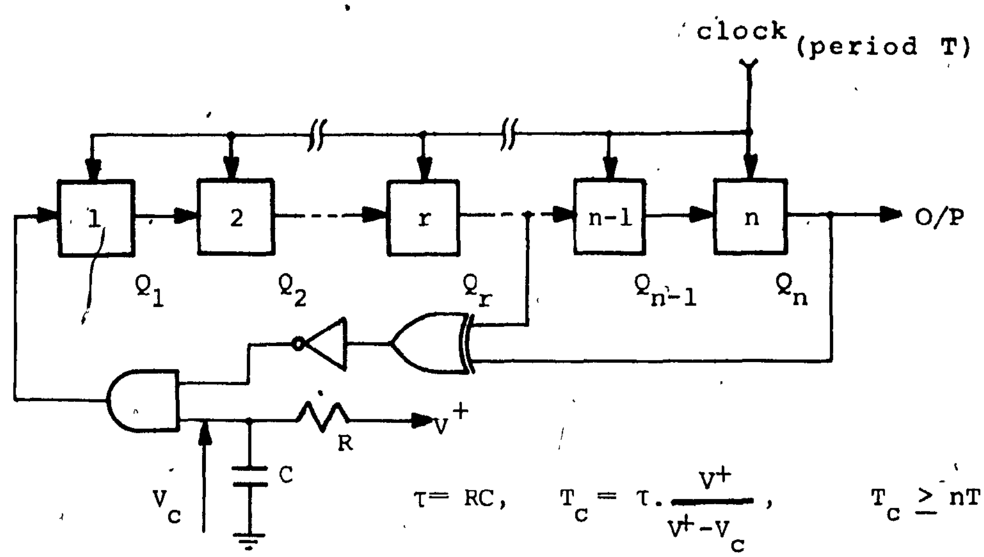


Fig.2.15 Configuration III: R and C are Chosen such that $T_c \geq nT$ Which Assures a Logic 1 at Input of Stage 1 during an All-Zero Start-Up.

Fig.2.13 is the simplest configuration in which the manual resetting of a switch at power-on instant would start the normal operation of the circuit. Figs.2.14 and 2.15 show automatic insertion of a logic 1 into the first stage of the shift-register. In Fig.2.15 the long time delay of RC network ensures that a logic 1 is always fed into the shift-register <u>at start-up</u> regardless of the initial state of shift-register. This action is based on the assumption that an all-zero condition would occur only at power-on instant, which is a valid assumption because once the maximal length sequence is started it never allows the shift-register to go into an all-zero state in normal operation (i.e., when there is no external interference during operation). In Fig.2.14 the NOR gate output is a logic 1 only when all its inputs are logic zero. The complexity of this circuit increases, however, with the number of stages in the shift-register.

In each of the above configurations the modulo-2 addition is performed using the outputs of stages that yield the maximum sequence length. Configuration II was adopted in the work reported here. The complete structure of the feedback shift-register of 31 stages is shown in Fig.2.16. Eight 4-input NOR gates and one 8-input AND gate were used to synthesize the function of a single 31-input NOR gate necessary for implementation. The maximal sequence length obtained from this circuit is $2^{31} - 1 \cong 2.20 \times 10^9$ bits, which when clocked at 500 kHz takes, approximately, 1 hour 11 minutes 35 seconds before repeating itself.

CMOS integrated circuits (ICs) were used in the implementation of PRBS generator. A simple RC low pass filter with various combinations of values of R and C for different cut-off frequencies, was used to filter the sequences. The filtered PRBS were tested for a number of properties. Their results were compared with those of gaussian noise samples generated by a conventional noise source. The CRO, Spectrum Analyzer and Probability Density Analyzer were used to observe displays of noise samples in

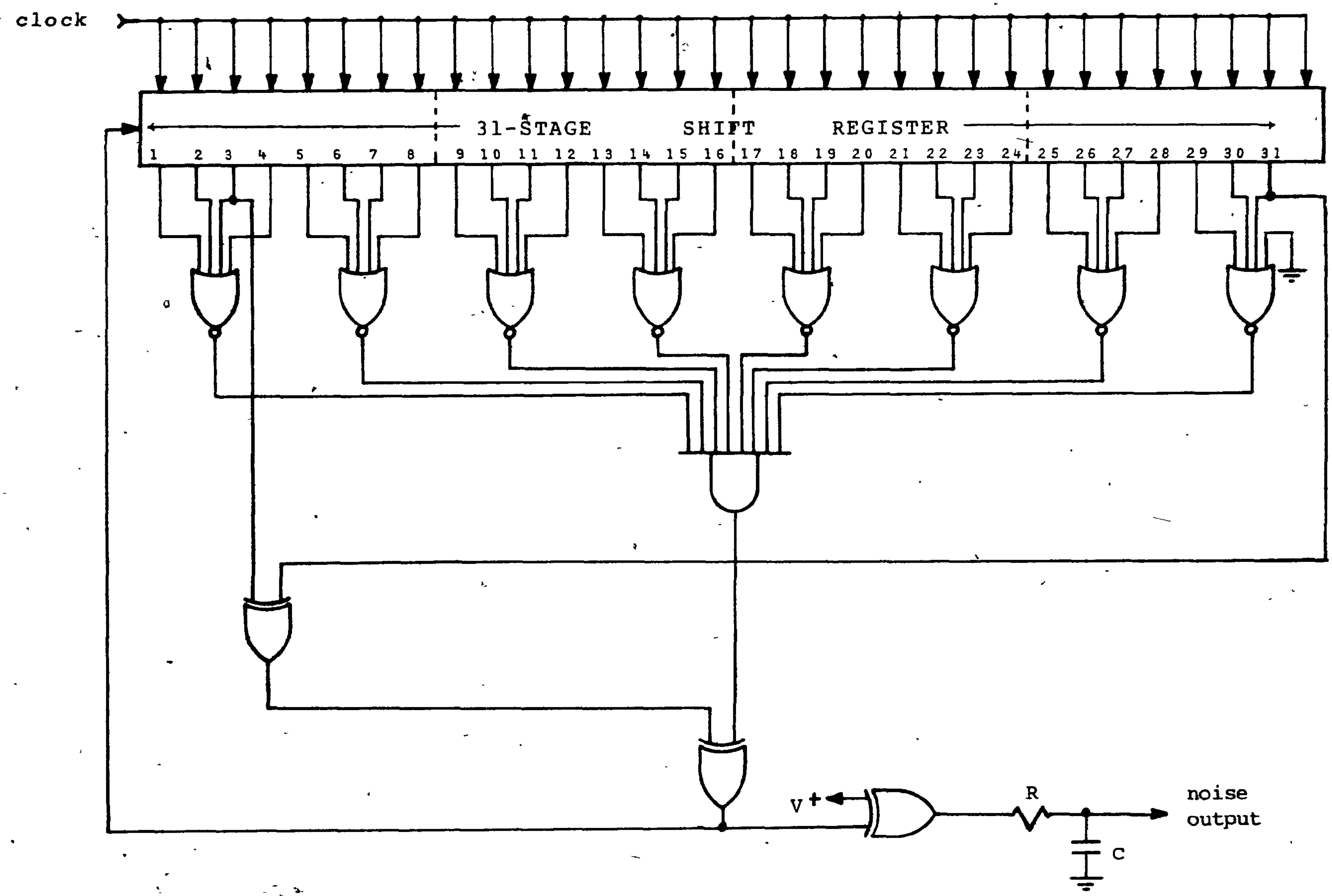
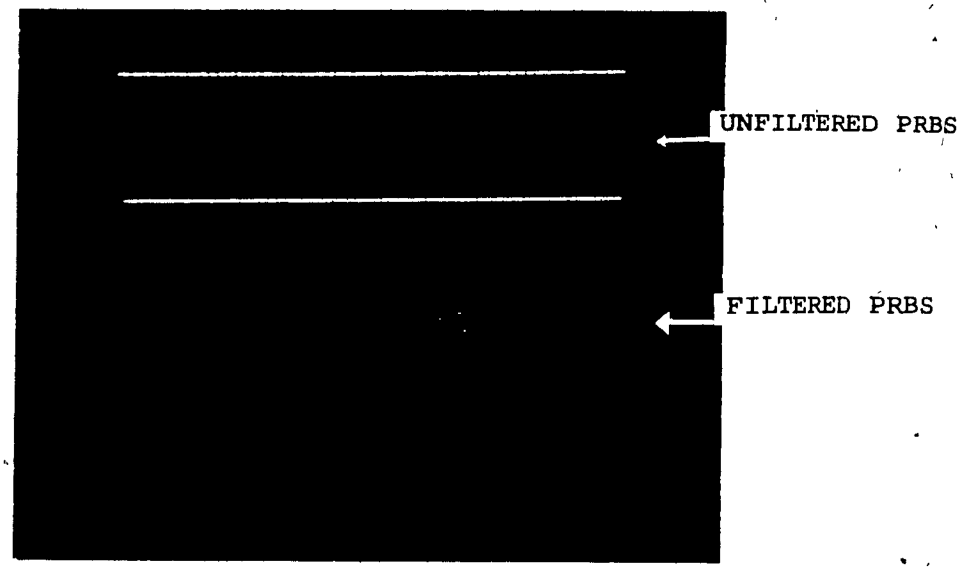


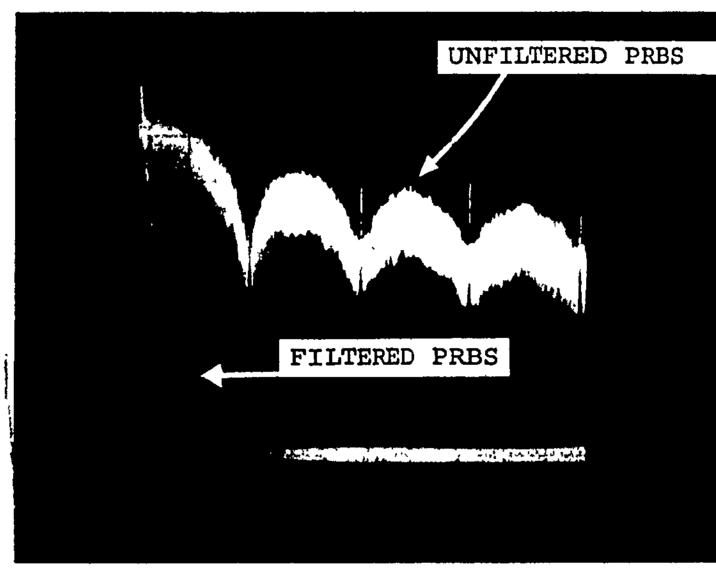
Fig.2.16 31-Stage Feedback Shift-Register Cascaded with a RC Low Pass Filter.

time-domain, frequency-domain and their statistical characteristics, respectively. Figures 2.17 through 2.27 show photographs of these displays.



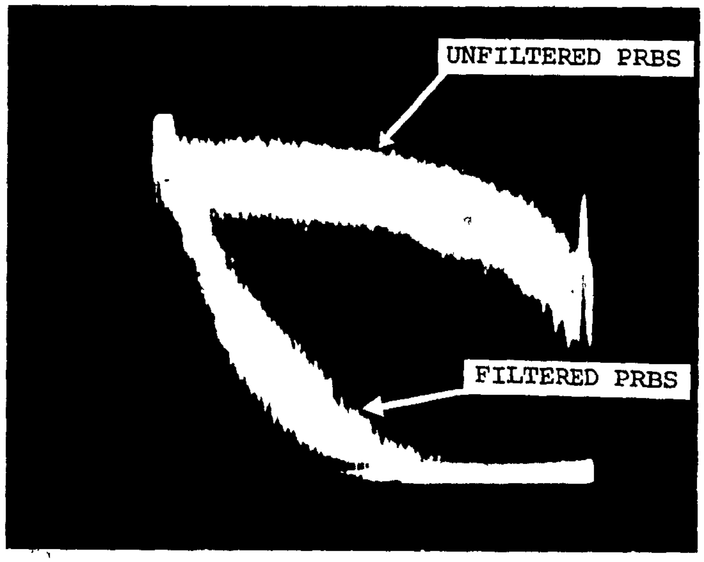
(H: 1ms/div , V: 2V/div & 1V/div)

Fig.2.17 Time-Domain Display of Unfiltered and Filtered PRBS. (Low Pass Filter: Simple RC, f_c = 25 kHz; f_b = 500 kb/s)



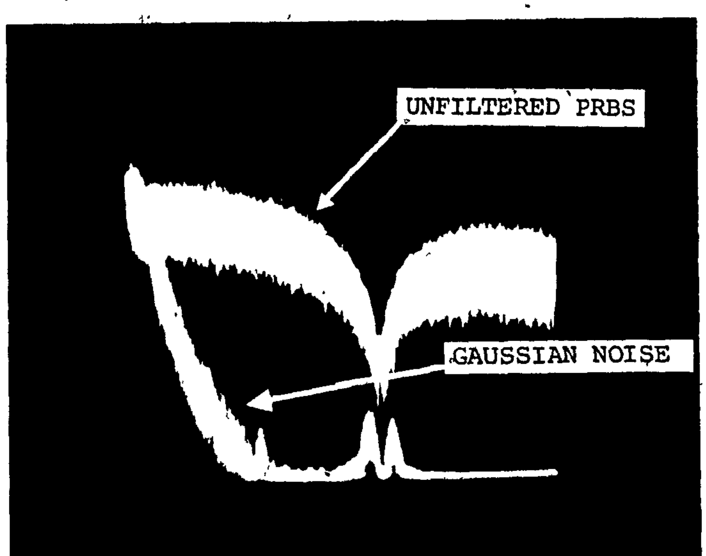
(H: 200kHz/div , V: 10dB/div)

Fig.2.18 Frequency-Domain Display of Unfiltered and Filtered PRBS Shown in Fig.2.17. (Note: Spikes are due to a Leakage of the Clock Signal)



(H: 50 kHz/div , V: 10dB/div)

Fig.2.19 The Same Power Spectra as in Fig.2.18 Except that Horizontal Scale is Expanded.

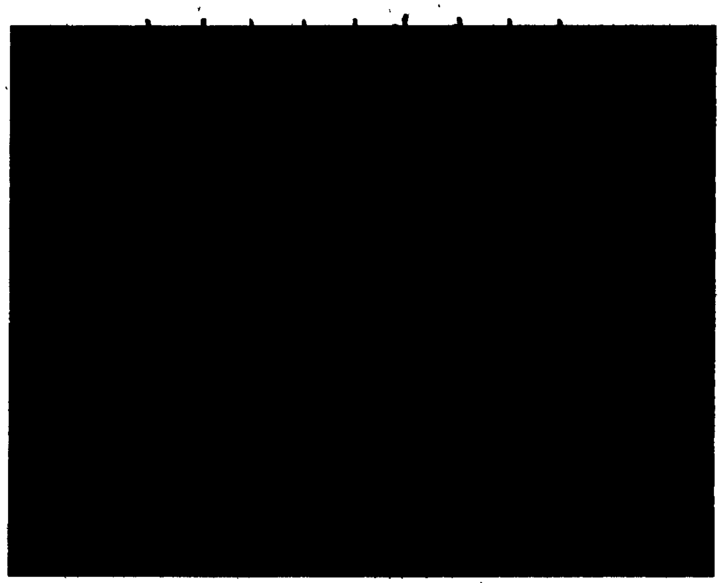


(H: 50kHz/div , V: 10dB/div)

Fig.2.20 Power Spectra of PRBS and Noise Samples from HP3722A Noise Generator.

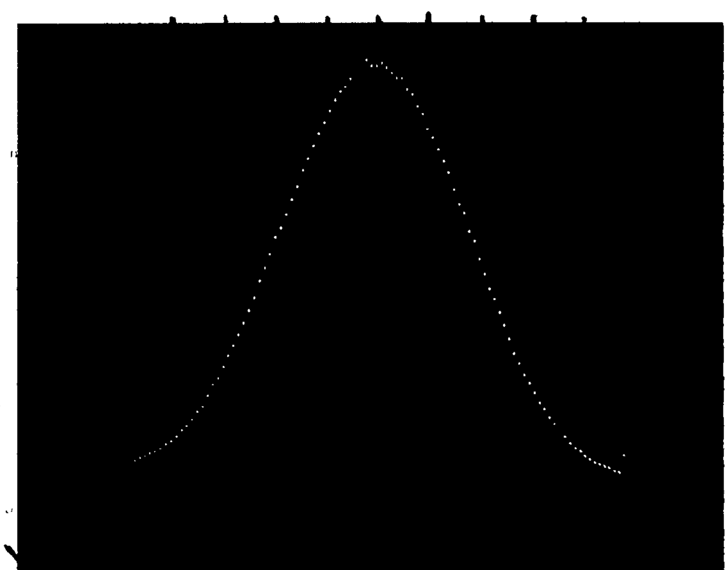
Fig.2.17 shows the unfiltered and filtered PRBS displayed on CRO. Their corresponding power spectra are shown in Figs.2.18 and 2.19. These displays, therefore, are the time-domain and frequency-domain representations of the same pair of signals. The PRBS were generated at 500 kb/s and filtered with a RC low pass filter of 25 kHz cut-off frequency, which corresponds to $\kappa = 20$ (as defined in Section 2.3.2). Fig.2.20 shows a display of the power spectra of gaussian noise generated by a conventional noise source (HP3722A) and of 300 kb/s PRBS generated by the same instrument. In comparing the Figs.2.19 and 2.20 it may be observed that the power spectrum of the filtered PRBS shown in Fig.2.19 has a shape similar to that of noise generated by the conventional source having a guaranteed gaussian noise characteristic.

The noise output of HP3722A unit was fed into HP3721A Correlator (cum p.d.f. analyzer) to observe the amplitude (voltage) p.d.f. as shown in Fig.2.21. As expected, this display shows a gaussian p.d.f., which was obtained after averaging 2^{13} noise samples. Fig.2.22 is a better display of the same signal when averaged over 2^{17} samples. It can be observed from these two displays that a smooth curve is obtained when the averaging is done on a large number of samples, as it was pointed out in Section 2.3.2. To detect any discrepancies between the two averaging values, the Correlator displays of Figs.2.21 and 2.22 were traced and



H: 0.1 V/div

Fig.2.21 P.d.f. of Noise Samples from HP3722A Noise Generator. (Averaged Over 2^{13} Samples)



H: 0.1 V/div

Fig.2.22 P.d.f. of the Same Signal as in Fig.2.21 Except for Averaging Over 2^{17} Samples.

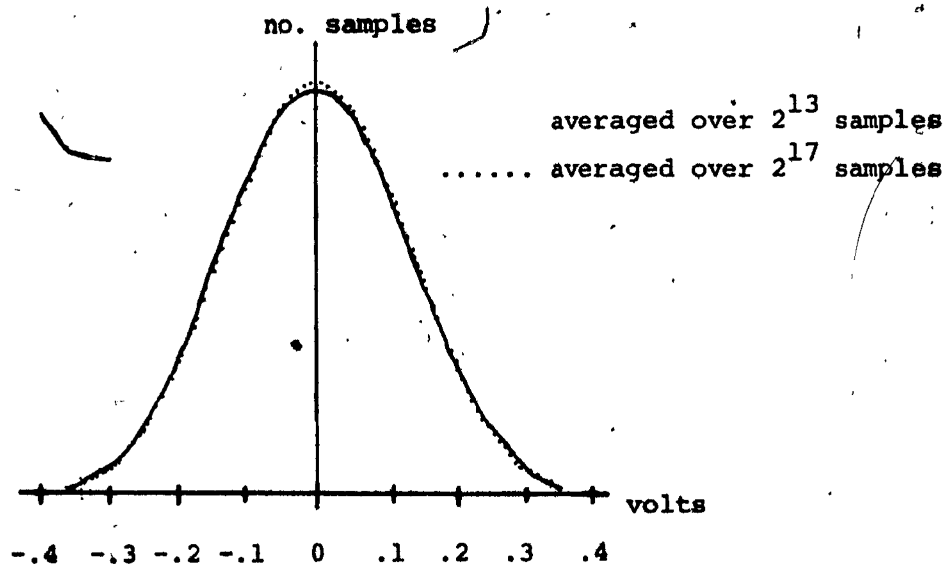


Fig.2.23 Superimposed Curves (Traced) of Figs.2.21 and 2.22.

superimposed on each other as shown in Fig.2.23. Only a minor difference between the two displays may be observed at peaks of the density functions. However, because of the longer time that the Correlator takes to average the larger number of samples a choice was made to average over 2^{15} samples (an in-between value) in the tests as explained next.

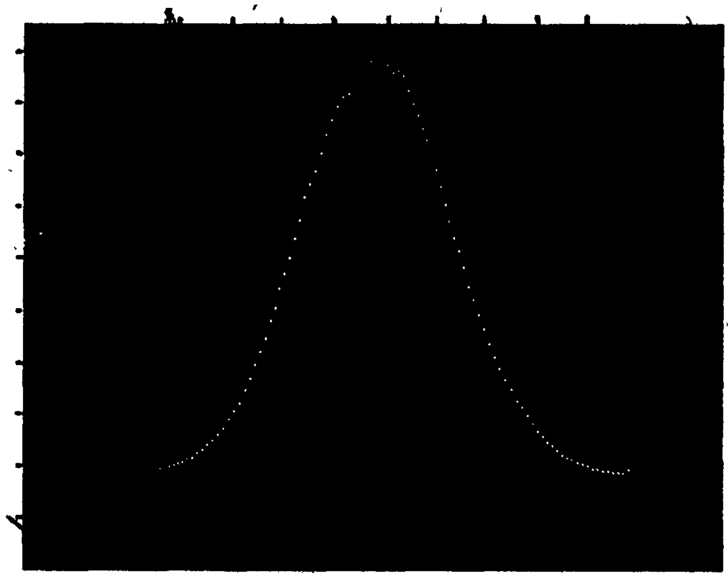


Fig.2.24 P.d.f. of Noise Samples from Prototype Noise Generator. (PRBS Length = $2^{31}-1$ bits; Clock: 500 kHz; LPF: RC, f_c = 23 kHz; Averaged Over 2^{15} Samples)

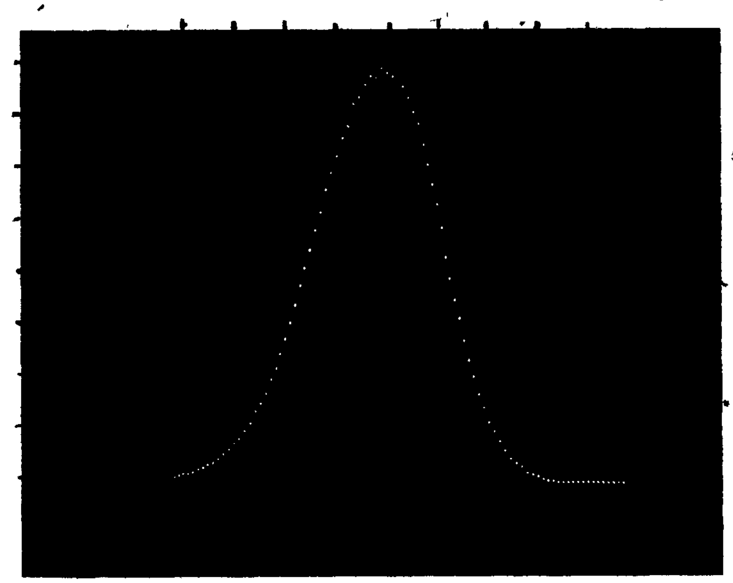


Fig.2.25 P.d.f. as in Fig.2.24 but f_c = 15 kHz.

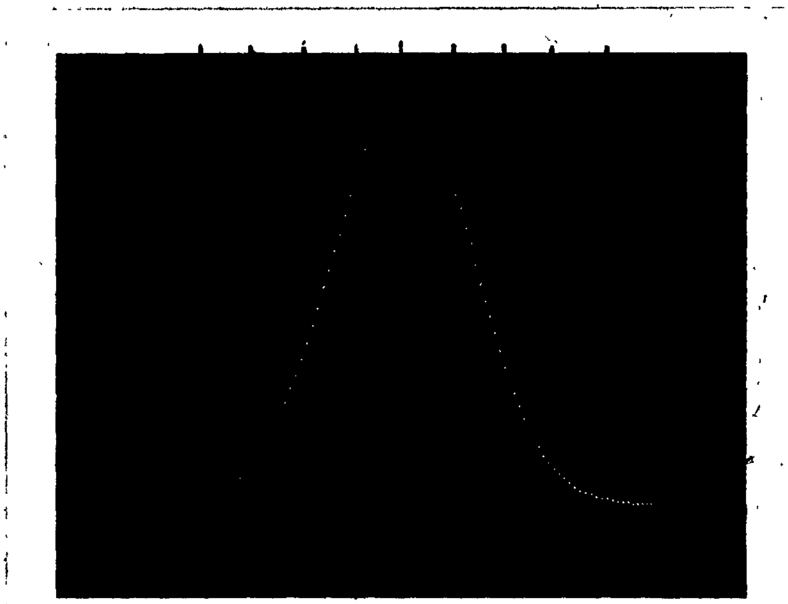
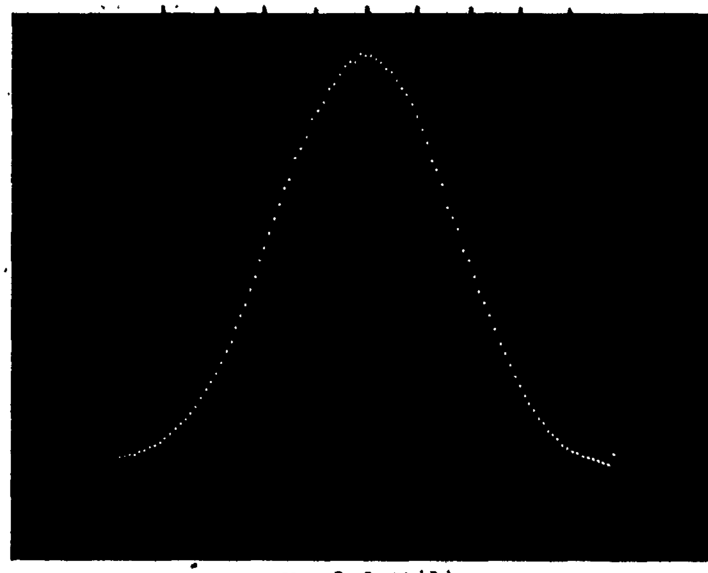
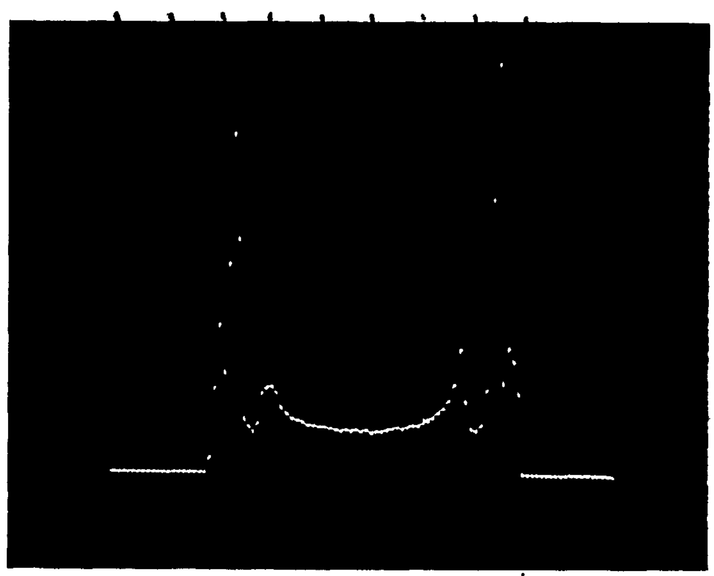


Fig.2.26 P.d.f. as in Fig.2.24 but f_c = 20 kHz.



H: 0.1 V/div

Fig.2.27 P.d.f. as in Fig.2.24 but f_c = 30 kHz.



H: 0.1 V/div

Fig.2.28 P.d.f. as in Fig.2.24 but with f_c = 300 kHz.

Figs.2.24 through 2.28 show amplitude density functions of filtered PRBS generated by the prototype noise generator for different values of cut-off frequencies of the RC LPF. It may be observed that around a 30 kHz cut-off frequency the amplitude p.d.f. of noise samples generated by the prototype unit approximates very closely that of samples generated by the conventional unit (HP3722A). The different displays were traced and superimposed on one another as

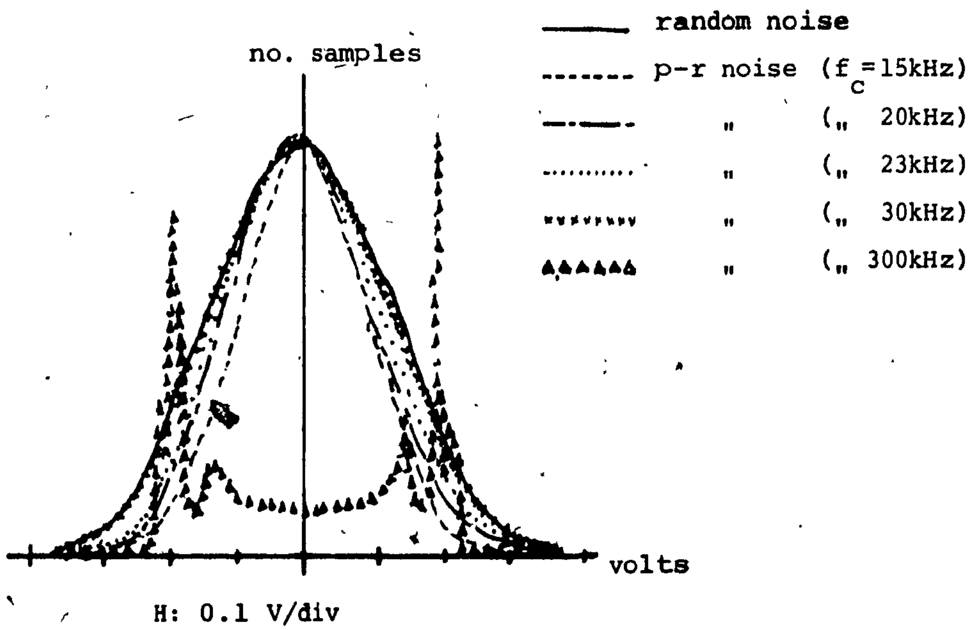


Fig.2.29 Superimposed Curves (Traced) of Figs.2.24 through 2.28.

shown in Fig.2.29. In this figure, a trace of the display in Fig.2.22 (gaussian p.d.f.) was re-drawn, as shown by the continuous line, to compare with other experimental results. It can now be seen how the p.d.f. of filtered PRBS approaches the gaussian curve as low pass filter cut-off frequency is increased from 15 kHz, but then completely deviates from the gaussian shape as the frequency is further increased beyond 30 kHz. Therefore, it is apparent that the 30 kHz cut-off frequency yields a good approximation to the gaussian when the PRBS is generated at 500 kb/s as depicted clearly in Fig.2.30. The autocorrelation functions of noise samples from both HP3722A and prototype units are shown in

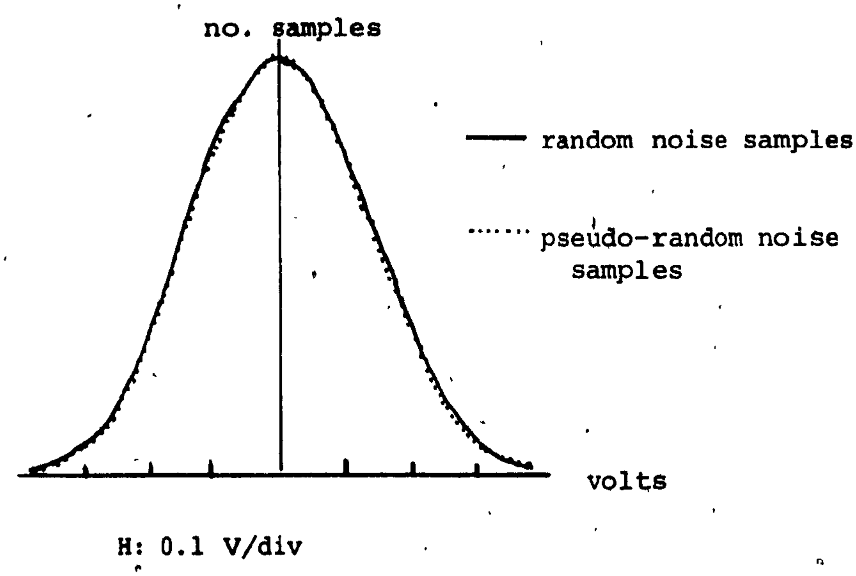
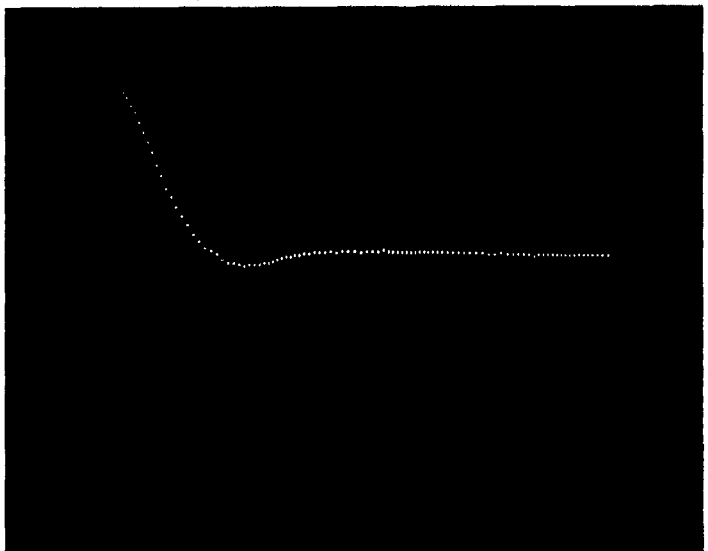


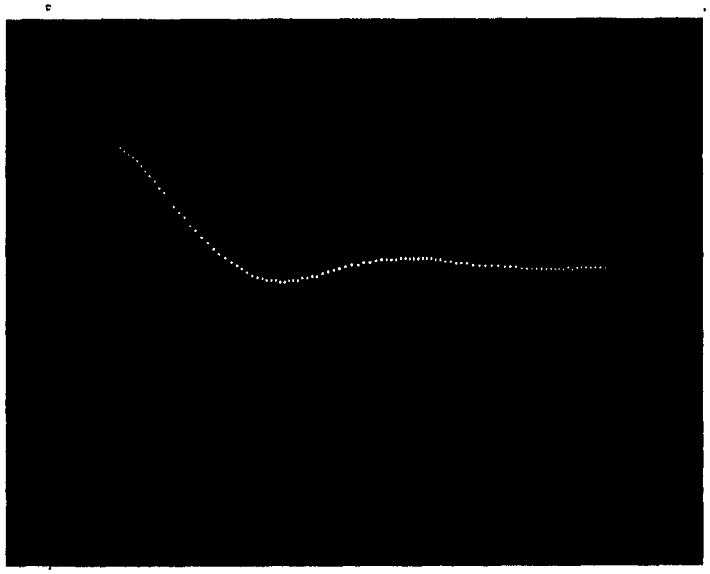
Fig.2.30 Superimposed Curves (Traced) of Figs.2.22 and 2.27.

Figs.2.31 and 2.32.



H: 1.0 μs betw' consecutive points

Fig.2.31 Autocorrelation Function of Noise Samples from HP3722A Noise Generator.

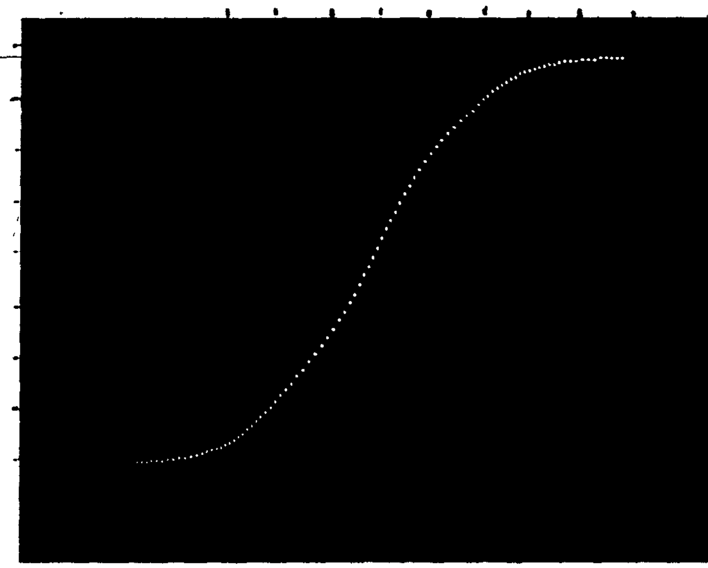


H: 1.0 μs betw' consecutive points

Fig.2.32 Autocorrelation Function of Noise Samples from Prototype Generator. PRBS Length = $2^{31}-1$ bits; LPF: RC, f_c = 23 kHz; f_b = 500 kb/s)

Finally, Figs.2.33 and 2.34 show the amplitude distribution functions of the two types of noise samples. Fig.2.35 was obtained by tracing Figs.2.33 and 2.34 so that their superposition would aid in comparing the two displays. The amplitude p.d.f. displays show that with a cut-off frequency between 23 kHz and 30 kHz of a simple RC low pass filter, the PRBS being generated at 500 kb/s, the noise samples at the filter output would approximate very closely a gaussian p.d.f. characteristic. This observation is in close agreement with the computer simulation results reported in the preceding section.

The rms voltage of the noise samples, whose p.d.f. is shown in Fig.2.27, was measured at 0.6 volts. This value of rms voltage is the standard deviation, σ, of this p.d.f. Therefore, in Fig.2.30 (superimposed curves of Figs.2.23 and 2.27) it is seen that the noise generated by the prototype as well as the conventional units are gaussian within 3σ.



H: 0.1 V/div

Fig.2.33 Probability Distribution Function of Noise Samples from HP3722A Noise Generator. (Averaged Over 2^{17} Samples)

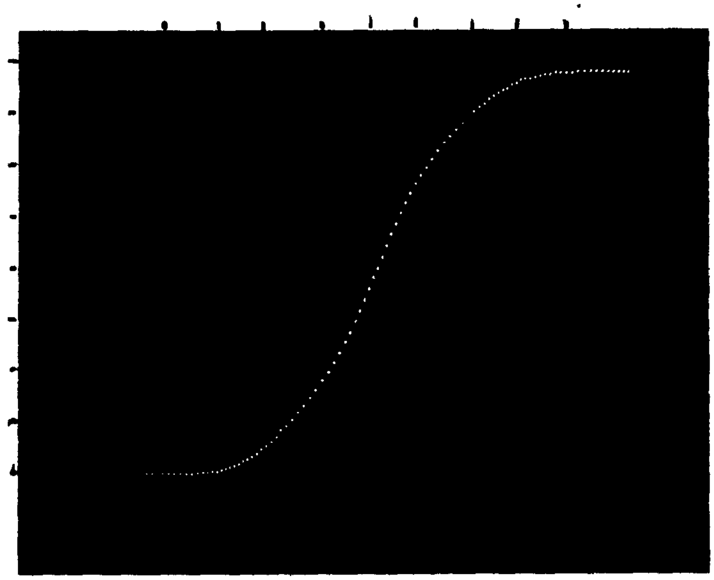


Fig.2.34 Probability Distribution Function of Noise Samples from Prototype Noise Generator. (Averaged Over 2^{15} Samples)

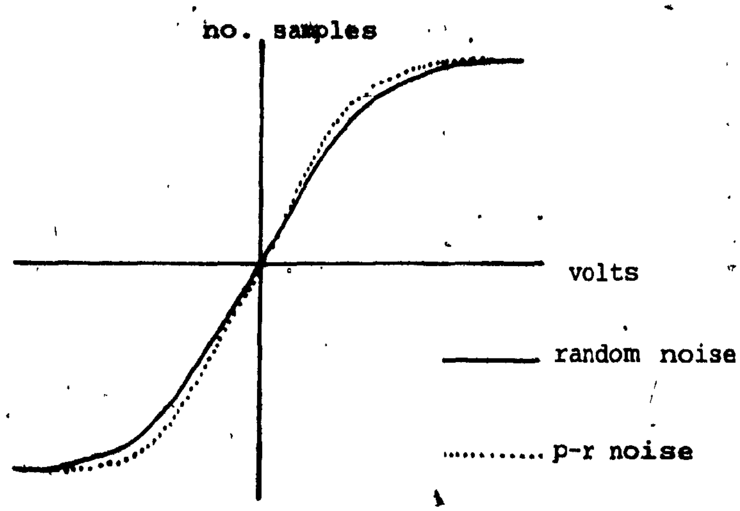


Fig.2.35—Superimposed Curves (Traced) of Figs.2.33 and 2.34.

In conclusion, it can be stated that in order to obtain gaussian noise samples from a pseudo-random sequence, the sequence may be filtered through an RC LPF whose cut-off frequency is kept around one-twentieth of the frequency of the clock that drives the PRBS generator.

The prototype gaussian noise source was evaluated in a baseband model of a digital transmission system. The results are reported in Chapter 4.

Chapter 3
TRANSVERSAL FILTERS

3.1 Introduction

The transversal filter (TF) dates back to 1935 when it was invented by Wiener and Lee [Lucky et al., 1968], but was first described in the literature by Kallman [1940]. To understand the fundamentals of TF, consider a linear,

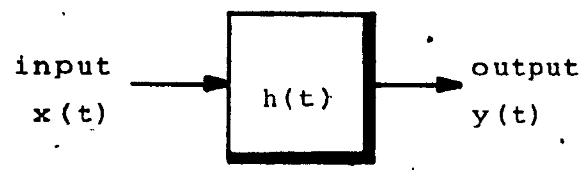


Fig.3.1 Block Diagram of a Linear, Passive, Time-Invariant Network.

passive, time-invariant network as shown in Fig.3.1. The output y(t) of the network in response to an input x(t) may be expressed in time-domain, using the convolution integral, as

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d \qquad (3.1)$$

where $h(\tau)$ is the impulse response of the network. Eqn.(3.1) may directly be implemented in time-domain if it would be possible to realize the device as shown in Fig.3.2.

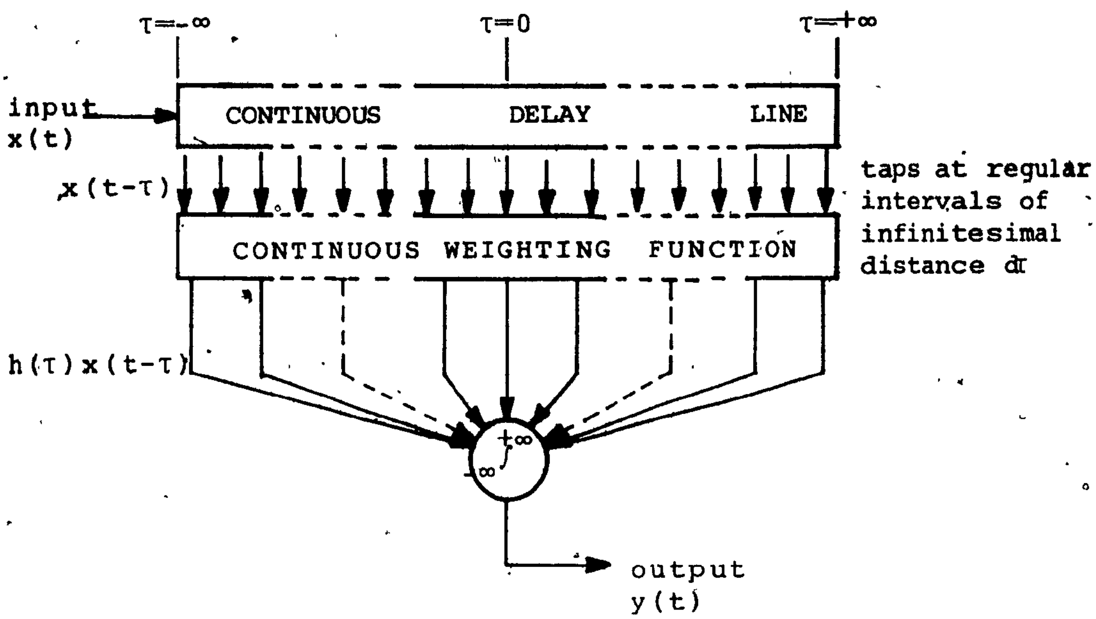


Fig.3.2 Time-Domain Model for Eqn.(3.1).

One of the most important characteristics of this device is that the signal x(t) propagates through the continuous delay line without being distorted, and the output signal y(t) is obtained by processing x(t) in the sense "transversal" to that of propagation. Such a device was termed by Kallman [1940] as an Ideal Transversal Filter.

In practice, however, the ideal TF is approximated by a "finite" length delay line tapped at "regular" intervals of 'T' (T is the duration of one symbol of the digital signal). This practical TF can easily be realized using the present day digital hardware in which a shift-register clocked at T sec. may be used in place of the continuous delay line and a discrete summing device instead of the integrator. Fig.3.3 shows a block diagram of a realizable TF. In view

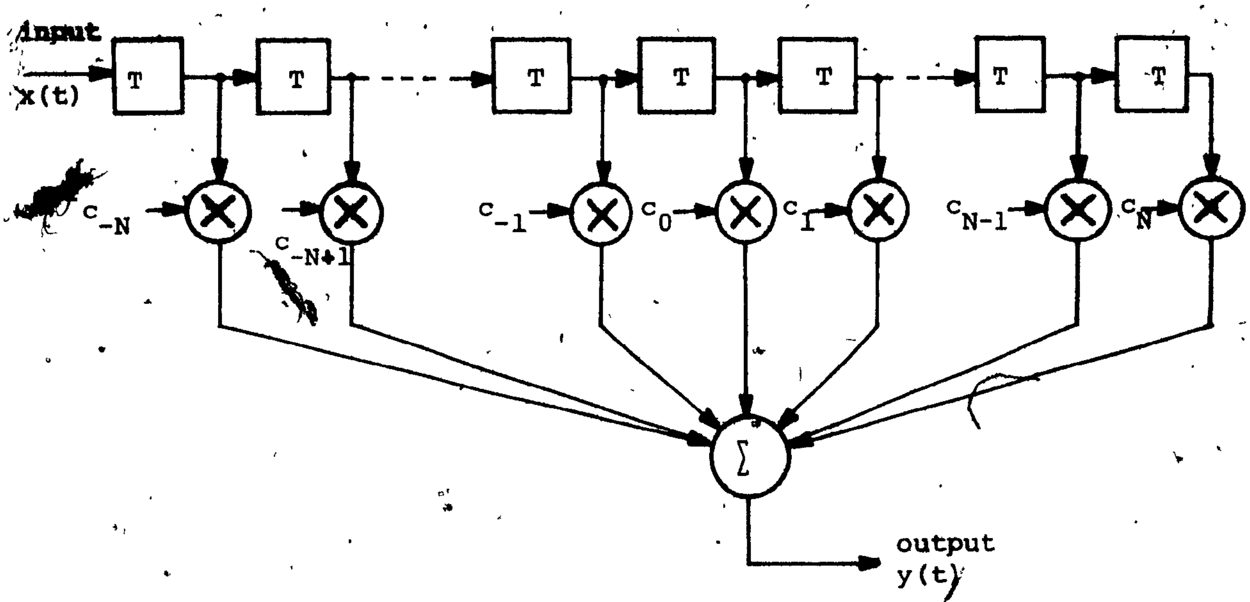


Fig.3.3 Block Diagram of a Practical Transversal Filter.

of the realization as shown in Fig.3.3, eqn.(3.1) can be modified as

$$y(t) = \Sigma_{j=-N}^{N} c_j \; x(t-jT) \tag{3.2}$$

where c_j are (2N+1) discrete weighting coefficients that replace the continuous weighting function in the ideal case. In modifying eqn.(3.1) into eqn.(3.2) errors are introduced to the transversal filter structure owing to,

i. limitation of the delay line in its length,

ii. replacement of the integral by a discrete summation, and

iii. replacement of continuous weighting function by a set of discrete weighting coefficients.

An extensive study on this approximation error was reported by Voelcker [1968]. In synthesizing TFs, computation of the weighting coefficients involves the major part of the design.

3.2 Computation of Weighting Coefficients

Weighting coefficients, c_j, often termed as tap-gains or tap-gain coefficients of the TF, may be computed in several different ways. The three most frequently used methods are,

i. Fourier series method,

ii. least squares method, and

iii. a method proposed by Mueller [1973].

In the Fourier series method [Huelsman, 1970], first, the Fourier coefficients of the desired filter transfer function are computed and are associated with the tap-gains. Least squares method [Carrefour et al., 1973] is one which minimizes the error between the specified transfer function and the actual filter transfer function in the least square sense, yielding the corresponding tap-gains. Both these methods demand specification of the filter transfer function. An elaborate description of these methods is given by de Cristofaro [1976]. Mueller's method of TF synthesis takes a different approach in computing the tap-gains. Some of the advantages of Mueller's method are,

i. filter transfer function need not be specified;

ii. coefficients c_j are optimum in the sense that transmitted energy in the excess Nyquist bandwidth is maximized;

iii. ISI is minimized.

Mueller's synthesis starts with the theory that the coefficients c_j are the samples of a truncated impulse response of an ideal Nyquist filter. Based on this method, Feher and de Cristofaro [1976] designed and evaluated a binary TF (BTF: a TF for binary signals) for application in satellite communications, which meets the stringent requirements of in-band to out-of-band energy ratio and ISI for single-channel-per-carrier (SCPC) transmission systems. In his M.Eng. thesis de Cristofaro [1976] presents a detailed description of this design.

3.3 Equalization Using Transversal Filters

In data transmission, certain pulse shapes are desirable to minimize ISI. Especially, the departure from a linear phase vs. frequency characteristic of channels distorts the received pulse, and causes ISI. When it is desirable to transmit over channels which may vary considerably in their amplitude- and phase-frequency characteristics, it may be necessary to apply corrective means or equalization to keep the pulse distortion within reasonable limits, without any degradation in system performance.

The TF offers great flexibility when it is necessary to select among or to adjust several attenuation and phase characteristics. It can be adjusted to have any transmission characteristic from a knowledge of the impulse response of the desired characteristic [Bennett and Davey, Ch.15, 1965]. This is done by adjusting the tap-gains such that the pulse crosses the time-axis exactly at each sampling point, except at its own sampling point where the pulse has the maximum amplitude. Fig. 3.[illegible] illustrates this procedure.

The continuous curve depicts a single data pulse that is distorted as a result of some undesirable characteristics within the system. Sampling instants are indicated along the time axis as t_{-3}, t_{-2}, t_{-1}, t_0, t_1,... The pulse has

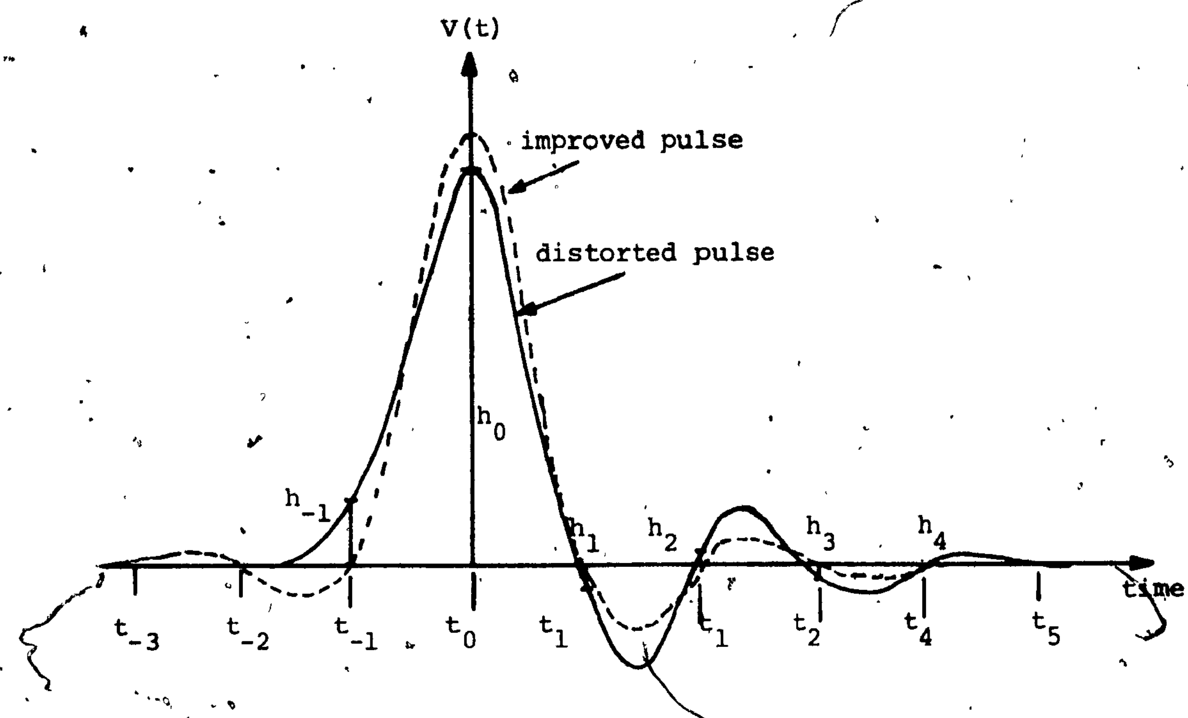


Fig.3.4 Improvement of a Pulse Response by the TF Action.

its peak amplitude h_0 at its own sampling instant t_0. However, at other sampling instances t_i ($i \neq 0$) the pulse has various amplitudes h_{-1}, h_1, etc., which, in turn, interfere with all other pulses (previous to and following the present pulse) at their respective sampling instants. If this distorted pulse were to pass through a TF (analog), the tap-gains could be adjusted so that the values h_{-1}, h_1, etc., can be made to approach zero. As is known, it is these interfering non-zero amplitudes h_{-1}, h_1, etc., that give rise to what is called ISI when a random data pattern replaces the single pulse. The adjustment of the tap-gains correspondingly yields the maximum eye opening during the random pulse train. The dashed line in Fig.3.4 shows the pulse after passing through the TF. This method of minimizing the interference of one pulse with the neighbouring ones at their sampling points is termed "minimization of peak distortion" as proposed by Lucky [1965]. The other method of minimizing ISI, which later appeared in the literature, is termed "minimization of mean-square-error distortion" [Lucky and Rudin, 1967]. Before looking into the details of the above two methods a few basic equations for TFs must be established.

For a 2N+1 taps (tapped at T sec. intervals) TF (as equalizer) with gains c_{-N}, c_{-N+1}, ..., c_0, c_{N-1}, c_N, the impulse response of the equalizer is, using eqn.(3.2),

$$e(t) = \Sigma_{j=-N}^{N} c_j \; \delta(t-jT) \qquad (3.3)$$

and in the frequency domain it takes the form**

$$E(\omega) = \Sigma_{i=-N}^{N} c_i \; e^{-j\omega iT} . \qquad (3.4)$$

If this TF is connected in tandem with a separate system whose impulse response is x(t), the impulse response of the overall equalized system x(t)*e(t), i.e., x(t) convolves with e(t), becomes,

---

** The variable j in eqn.(3.3) has been replaced by i to avoid ambiguity with j ($= \sqrt{-1}$) in eqn.(3.4).

$$h(t) = \Sigma_{j=-N}^{N} c_i \ x(t-jT) \tag{3.5}$$

At sampling times $(nT+t_0)$, then,

$$h(nT+t_0) = \Sigma_{j=-N}^{N} c_j \ x[(n-j)T+t_0] \tag{3.6}$$

or

$$h_n = \Sigma_{j=-N}^{N} c_j \ x_{n-j} \tag{3.7}$$

Eqn.(3.7) shows that the output sample sequence is formed from the input sequence by polynomial multiplication.

3.3.1 Minimization of Peak Distortion

Assume that a binary signal $a_k = \pm 1$ is transmitted. The received signal at sampling time $(kT+t_0)$, where t_0 is the time delay in its passage through the channel, can be written in the form,

$$y_k = h_0 a_k + \Sigma_{n \neq k} \ a_n h_{k-n} + \eta_k \tag{3.8}$$

where η_k denotes the noise sample at time $(kT+t_0)$. Ignoring the noise contribution for the moment, the received sample y_0 may be written* as,

---

* A prime on a summation indicates deletion of the zeroth term.

$$y_0 = h_0 a_0 + \Sigma_n' \, a_n h_{-n} \tag{3.9}$$

The maximum value of the summation in eqn.(3.9), indicating ISI, occurs when the data sequence $\{a_n\}$ is transmitted which utilizes for each symbol a_n the maximum signal level of the same algebraic sign as h_{-n} [Lucky et al., 1968]. Therefore,

$$y_0 = h_0[a_0 + (1/h_0) \, \Sigma_n' \, a_n |h_n|] \tag{3.10}$$

Since a_0 is the wanted term, and the rest is the ISI contribution, an interference criterion proportional to the second term called "peak distortion, D," may be formed as,

$$D = (1/h_0) \, \Sigma_n' \, |h_n| \tag{3.11}$$

Assume that the input pulse to the TF will have an initial peak distortion D_0 and the reference sample x_0 will be normalized to unity. The aim, here, is to determine the tap-gains c_{-N}, ..., c_N for a (2N+1)-tap TF that minimizes the final peak distortion given in eqn.(3.11), where

$$h_n = \Sigma_{j=-N}^{N} \, c_j x_{n-j} \tag{3.12}$$

Constraining the central sample h_0 to unity, in order to take the arbitrary gain factor involved in the TF into account, one may write (replacing n in eqn.(3.12) by 0)

$$h_0 = 1 = \Sigma_{j=-N}^{N} c_j x_{-j} \tag{3.13}$$

Center tap c_0 can be used to satisfy this constraint. Thus, because the reference sample, x_0, is normalized to unity,

$$c_0 = 1 - \Sigma'^{N}_{j=-N} c_j x_{-j} \tag{3.14}$$

Substituting for c_0, eqn.(3.12) then becomes,

$$h_n = \Sigma'^{N}_{j=-N} c_j (x_{n-j} - x_n x_{-j}) + x_n \tag{3.15}$$

Now, because $h_0 = 1$, the output pulse distortion given by eqn.(3.11) becomes,

$$D = \Sigma'^{+\infty}_{n=-\infty} \left| \Sigma'^{N}_{j=-N} c_j (x_{n-j} - x_n x_{-j}) + x_n \right| \tag{3.16}$$

Thus, c_0 being eliminated to satisfy the constraint on h_0, what is left now is to minimize D in eqn.(3.16) over the 2N variables c_j, $|j| \leq N$, $j \neq 0$. Peak distortion, D, is a continuous, piecewise linear function of the tap-gains c_j [Lucky, 1965]. Eqn.(3.16) may, then, be rewritten in the form,

$$D = \Sigma'^{N}_{j=-N} c_j [\ \Sigma'^{+\infty}_{n=-\infty} (x_{n-j} - x_n x_{-j})\ \mathrm{sgn}(h_n)] + [\ \Sigma'^{+\infty}_{n=-\infty} x_n\ \mathrm{sgn}(h_n)] \tag{3.17}$$

where
$$\mathrm{sgn}(h_n) = \begin{cases} +1 & h_n \geq 0 \\ -1 & h_n < 0 \end{cases} \tag{3.18}$$

It has been shown [Lucky, 1965] that in the above equation the coefficients c_j are constant over certain regions of the 2N-dimensional space of definition for $\{c_j\}$. Breakpoints where the coefficients assume new values occur whenever an output sample h_n changes sign. A minimum of D cannot occur between breakpoints where the function is linear; thus, at least one value h_{k1} must be zero at the minimum. The equation $h_{k1} = 0$ may be used to eliminate one of the variables c_j. The reduced equation of the same piecewise linear form, requires at least one or more output samples $h_{k2} = 0$, etc. Thus, the setting of a (2N+1)-tap TF (as an equalizer) that minimizes peak output pulse distortion forces 2N zeros at sampling points of the output pulse response.

It must also be noted that,

(i) if $D_0 < 1$, then the minimum distortion D must occur for those 2N tap-gain settings which cause $h_k = 0$, $|k| \leq N$, $k \neq 0$;

(ii) the peak distortion given by eqn.(3.16) is a convex function of the 2N variables c_n, $|n| \leq N$, $n \neq 0$ [Lucky, 1965].

3.3.2 Minimization of Mean-Square Distortion

The mean-square distortion of the output response may be defined as,

$$\varepsilon^2 = (1/h_0^2)\, \Sigma_n'\, h_n^2 \tag{3.19}$$

To minimize ε as a function of tap-gain settings h_0 may be constrained by means of a Lagrange multiplier. Thus, one looks for the point where $\partial V/\partial\lambda_j = 0$ for $j = -N,\ldots,N$, where

$$V = \Sigma_{n=-\infty}^{+\infty}\, h_n^2 + \lambda h_0 \tag{3.20}$$

Because

$$h_n = \Sigma_{j=-N}^{N}\, c_j x_{n-j} \tag{3.21}$$

then,

$$\partial V/\partial c_j = \Sigma_{n=-\infty}^{+\infty}\, 2h_n x_{n-j} + \lambda x_{-j} = 0 \tag{3.22}$$

The overall gain h_0 can be adjusted by proper scaling. It is, therefore, convenient to solve the set of equations,

$$\Sigma_{n=-\infty}^{+\infty}\, h_n x_{n-j} = x_{-j} \tag{3.23}$$

where $j = -N,\ldots,N$. Substituting for h_n given by the eqn.(3.21),

$$\Sigma_{k=-N}^{N}\, c_k b_{j-k} = x_{-j} \tag{3.24}$$

where $j = -N, \ldots, N$, and

$$b_k = \Sigma_{n=-\infty}^{+\infty} x_n x_{n-k} \tag{3.25}$$

The best tap-gain coefficient in a mean-square sense are, therefore, obtained by solving simultaneously the (2N+1) linear equations (3.24), and subsequently scaling the values obtained to achieve the proper pulse height.

3.4 Predistortion of Digital Signals Using Transversal Filters

In foregoing sections the transversal filter was presented as a filter that is used at the receiving end of a transmission system, in view of equalizing the undesired channel characteristics. It must not, however, be overlooked that this versatile device can also be used at the transmitting end either as a transmit filter by itself (in lowpass or bandpass modes) [Spilker, Ch.13, 1977] or as an equalizer for some other conventional transmit filters. The binary TF reported by Feher and de Cristofaro [1976] yielded satisfactory results in compliance with single-channel-per-carrier (SCPC) system requirements regarding bandlimitation and ISI of the baseband signal. However, although a BTF would be self-adaptive to a wide range of data rates the above mentioned filter is deprived of this capability because of the cascaded smoothing filter at the last stage [de Cristofaro, p.87, 1976]. Nevertheless, in many cases the BTF is designed for a specific data rate and adaptability is not an important factor. In this hybrid filter design emphasis was laid on the BTF; the second-order Butterworth LPF cascaded to it is merely present to smooth out the output pulse and to increase the in-band to out-of-band energy ratio of the signal in the intended application.

The predistortion technique proposed in this thesis makes use of a TF in a similar method to the above mentioned BTF, but in reverse to its design approach. This is also a cascaded transversal/analog active filter. The proposed technique has proved to be much simpler and cost effective in implementation compared to the former. Moreover, easy field alignment can be effected should such a need arise.

This technique can simply be stated as follows:

(i) Design a conventional analog lowpass filter to meet the requirement on bandlimitation; there is no need to undertake a complex phase equalization of this analog filter.

(ii) Design a transversal filter and cascade it in front of the analog filter; adjust the tap-gains of TF to remove ISI caused by the analog filter.

Part (i) can be easily implemented by the well known conventional filter synthesis. A shift-register, a resistive network and an operational amplifier connected in tandem would solve part (ii) of the design procedure. The TF, thus, acts as a distorting device on the digital signal before it passes through the analog bandlimiting filter. This distortion is done in such a way as to oppose the distortion caused on the signal by the analog filter. The tap-gain coefficients of the TF can be obtained either manually using an oscilloscope or through an iterative computation method using a general purpose digital computer. In the latter case, either peak distortion or mean-square distortion criteria as discussed in the earlier sections can be utilized to compute the tap-gain coefficients. In both computation methods, however, the impulse* response of the analog low pass filter must be known. These methods are discussed in detail in Chapter 4, where the design and evaluation of a cascaded transversal/active analog lowpass filter are presented.

---

* A pulse response is sufficient in practice.

Chapter 4

DESIGN OF A CASCADED DIGITAL/ANALOG FILTER

4.1 Introduction

Analog filters have been widely used in digital communication systems to limit signals to the desired band of frequencies. These types of filters have evolved over many decades and their syntheses are well explored and described in the literature. Exhaustive analyses, including those done using digital computers, are readily available [Szentirmai, 1973]. Given the specific transfer characteristics, computer programs have been written to optimize the filter parameters that yield the precise function of the device [Antoniou, 1979]. For fast implementation one can simply refer to tables and/or graphs that specify component (passive) values necessary to approximate the desired characteristics [Johnson et al., 1980]. Over the past few years different manufacturers [Frequency Devices, 1979] have introduced to the market miniature modules of active analog filters that meet the requirements of different standard specifications such as those of CCITT.

When used for bandlimiting digital signals, these types of filters, however, give rise to ISI, a distortion not associated with analog signals. ISI is caused mostly by undesired group delay characteristics of analog filters. Sections of allpass networks can be used with these filters to flatten the otherwise curved group-delay characteristics within the band of frequencies concerned. Yet, the number of sections that may be necessary to achieve the desired degree of such equalization may become prohibitively large, and, thus, the composite filter may become too bulky and costly.

The design procedure proposed in this thesis simplifies matters to a great extent. It utilizes the readily available active analog filter syntheses for bandlimiting purpose, but equalizes the filtered signal by a predistortion process prior to filtering. The predistorting device used is a simple binary transversal filter. The latter has the added advantage of increasing the in-band to out-of-band energy ratio of the signal owing to its x/sin(x) spectral shape, thus, functioning as a whitening filter. The binary transversal filter, being a digital device, is readily compatible with the input NRZ signal, a common signal format in digital transmission systems.

4.2 Active* Filters for Bandlimiting the Digital Signals

Development of IC technology has contributed much to the advancement of filter design techniques. Usage of operational amplifiers as an isolating as well as a gain device has improved and simplified the synthesis of analog filters not only in performance but also in reducing size and cost comparatively. There are basically four types of active filters: (i) active filters with lumped resistor-capacitor (RC) elements, (ii) active filters with distributed RC elements, (iii) active N-path filters, and (iv) digital filters. The first type of active filters is more commonly known as "active RC filters" [Temes and Mitra, 1973].

Active RC filters offer a flexible solution in the design of bandlimiting filters for digital signals, in that passive components, viz., capacitors and resistors, can be chosen arbitrarily to obtain the desired function. The useful frequency range of active filters is determined primarily by the active elements being used, usually IC amplifiers. One drawback, however, in constructing these filters with discrete passive components is the necessity for individual "tuning" of component values. The reason being that it is almost impossible to obtain passive components with the exact values as computed in the design

---

* Active filters, as opposed to passive ones, are those that contain active devices, and they require power to operate.

synthesis. Computed values, therefore, must be rounded off to the nearest nominal values available. Also, the actual values of available components may differ from the nominal depending on the degree of tolerance as marked by the manufacturer. These two problems can, however, be almost eliminated if the entire filter is fabricated completely in IC form on a single silicon chip. This method also eliminates the problem of drift in component values during changes in ambient temperature. A limitation to this type of miniature filter is, obviously, the cost involved in the manufacturing process when produced on a small scale. However, in cases where the drawbacks mentioned are not of major consequence, active RC filters implemented with discrete components still offer the filter designer a simple and effective solution in signal processing.

Active filters can be designed either in direct or cascade form. In the direct form, the prescribed transfer function is realized directly as one section, whereas in the cascade form it is realized as a cascade of lower order sections. Cascade realization methods normally make use of active second-order filter sections and passive first-order filter sections. The active biquadratic sections are usually amplifier-based realizations. The cascaded form is adopted here because of its simplicity of implementation.

An eighth-order Butterworth low pass filter was designed and constructed in the laboratory to be used as the bandlimiting filter for digital signal. This high-order filter has a satisfactory steep roll-off of stop-band frequencies. The design procedure is very simple and widely known by filter designers, but is described in Appendix B for reference. Alternative procedures may also be found in text books [Johnson et al., 1980; Bruton, 1980]. The 8^{th}-order filter was tuned to have a 3 dB cut-off frequency of 16 kHz so that it may be used to bandlimit a 32 kb/s NRZ-baseband signal. The filter configuration is shown in

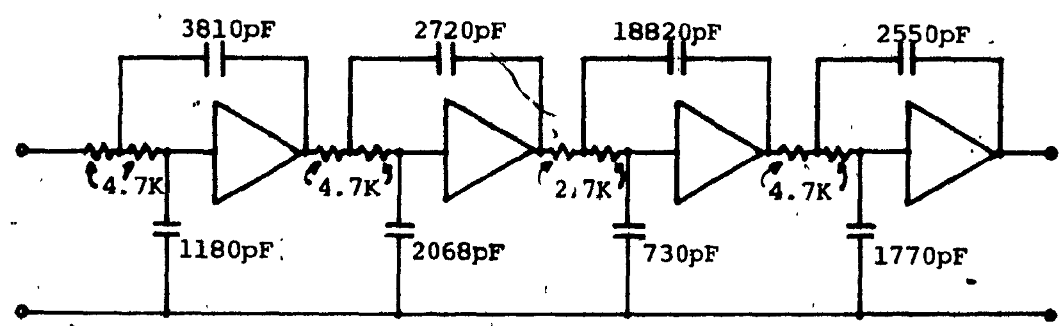


Fig.4.1 8^{th}-Order Butterworth LPF with Tuned Components.

Fig.4.1. In synthesizing the analog filter (AF) the gain of each amplifier was assumed to be unity. However, in a practical situation variations in the power supply may adversely affect the parameters in the operational amplifiers such as μA741 type, causing their gain setting

to move away from unity. To alleviate such discrepancies each unity-gain amplifier was realized by using two

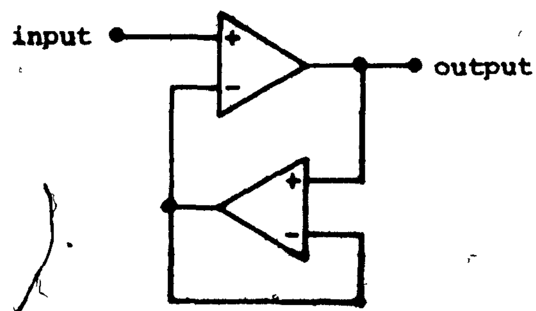


Fig.4.2 Configuration of Unity-Gain Amplifier Used in the Filter.

operational amplifiers connected as shown in Fig.4.2. This method was adopted based on results of the work done by Natarajan [1979]. The MC1458 is an 8-pin dual OA manufactured by Motorola Inc., which does not require any external compensating network. Therefore, four such IC packages were used to realize the unity-gain amplifiers.

The amplitude and phase characteristics of the 8^{th}-order LPF were measured against the frequency. The results are shown in Fig.4.3. The amplitude response is flat within 0 to -0.5 dB up to 14 kHz and rolls off sharply with -3 dB level at 16 kHz and at 48 dB/octave roll-off rate as intended.

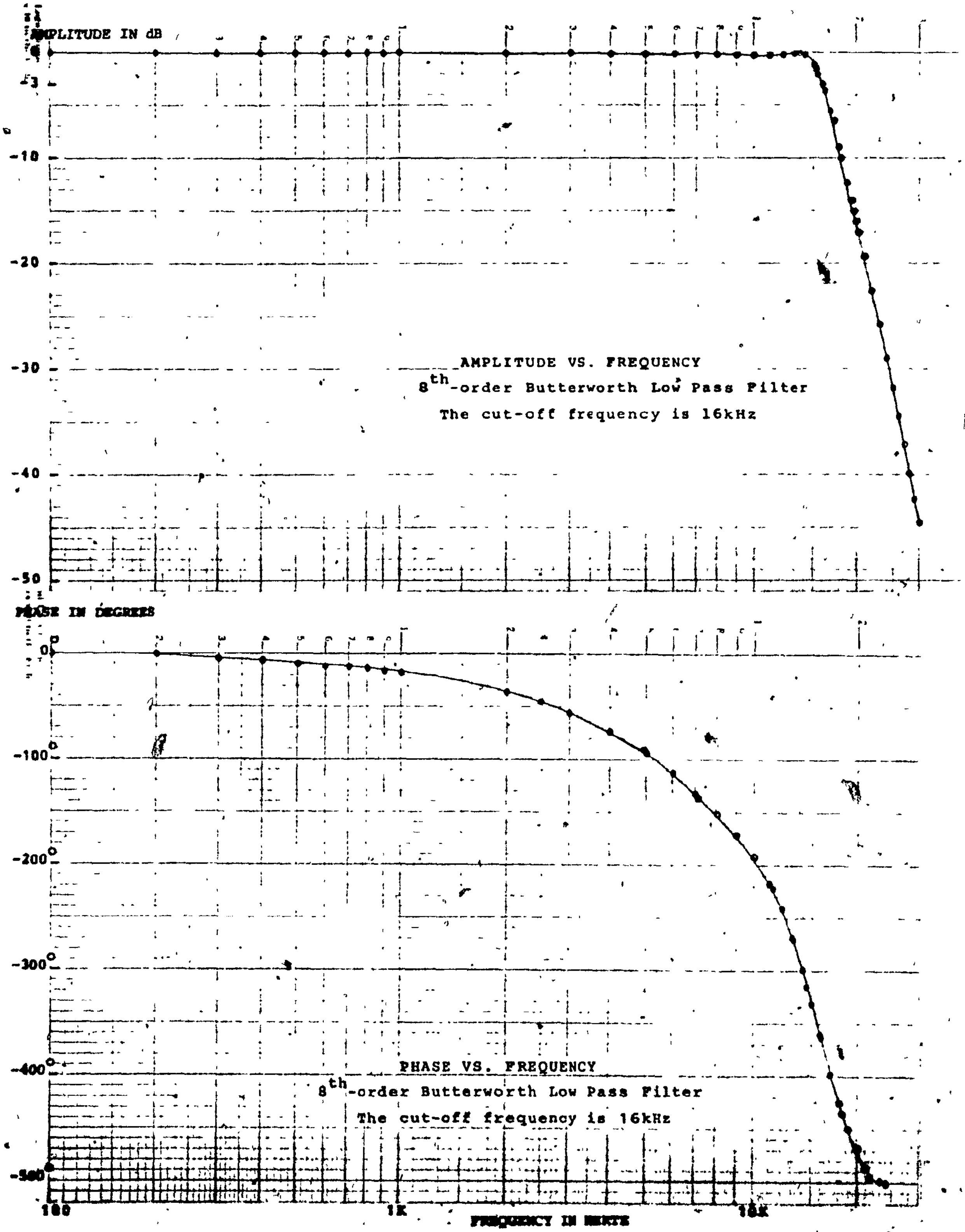


Fig.4.3 Gain and Phase vs. Frequency Characteristics of the 8^{th}-Order LPF(AF).

The analog filter so designed with cut-off frequency at 16 kHz was tested using the instrumentation set-up as shown

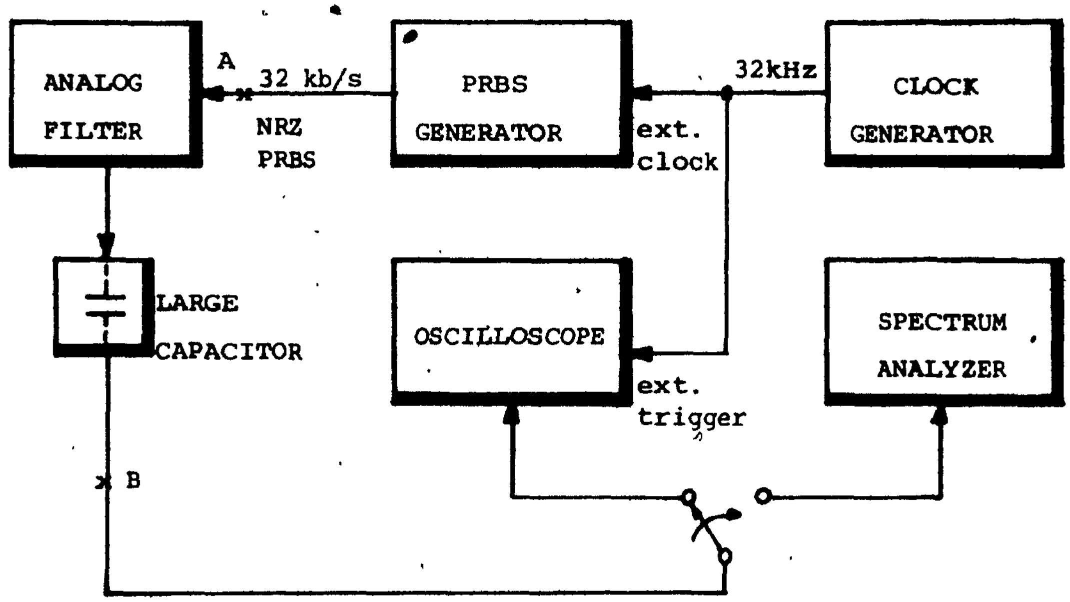
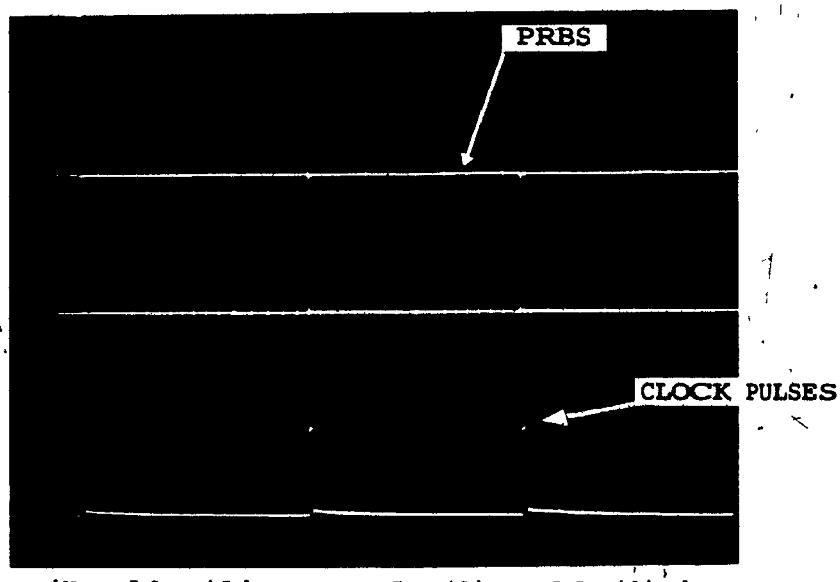


Fig.4.4 Instrumentation Set-Up for Testing the AF.

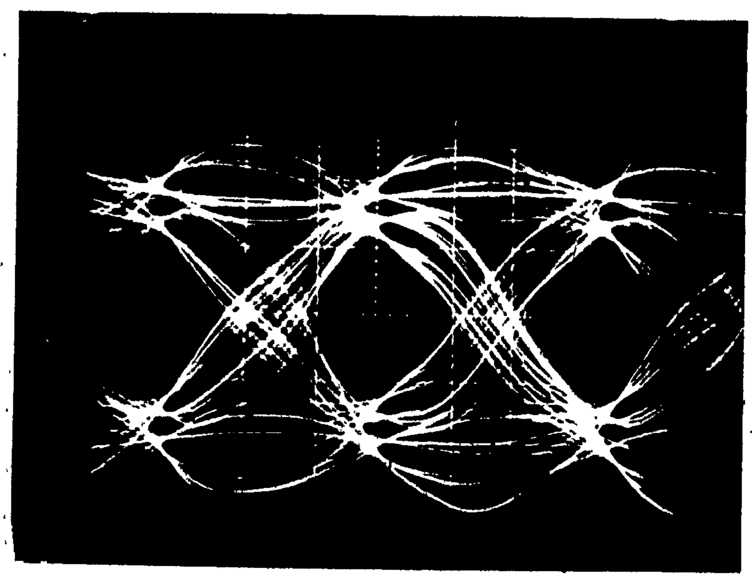
in Fig.4.4. Figs.4.5 and 4.6 show photographs of the random NRZ signal unfiltered and filtered as seen on CRO at points A and B of Fig.4.4, respectively.

Fig.4.7 shows the power spectra of the signal (superimposed) at points A and B as seen on the spectrum analyzer. Although the spectrum reveals that the signal has been well bandlimited the filter has, however, introduced the time-domain distortion ISI to the signal, which in effect has caused the "eye" to close somewhat at its probable sampling point. The peak degradation in the eye opening due to ISI is about 4.6 dB. This particular signal shape with heavy ISI is almost useless in a practical digital transmission



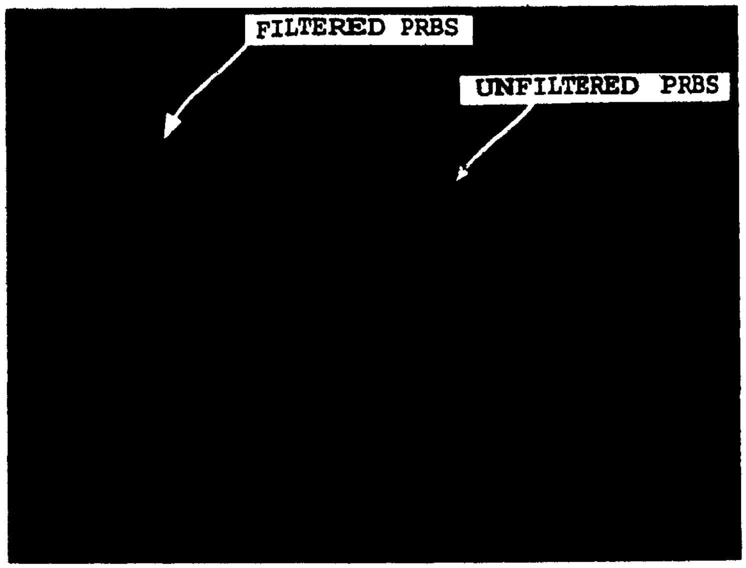
(H: 10µs/div , V: 1V/div &>1V/div)

Fig.4.5 PRBS at 32 kb/s and Clock Pulses at 32 kHz (Lower Trace).



(H: 10µs/div , V: 1V/div)

Fig.4.6 The Eye Diagram of PRBS of Fig.4.5 When Filtered with the AF (f_c = 16 kHz).



(H: 10kHz/div , V: 10dB/div)

Fig.4.7 Power Spectra of Filtered and Unfiltered PRBS Signal Shown in Figs.4.5 & 4.6.

system, because the channel noise added on to the signal during its passage would cause an intolerable error rate in the detection process at the receiver. This fact is reflected clearly in the bit error rate measurements, as depicted by curve 4 in Fig.4.26.

The major contribution of this thesis is to show how such a distortion in digital signals may be completely avoided through a predistortion process prior to filtering. The next section describes the design procedure of a predistorter that could be used in tandem with a simple analog type of bandlimiting filter described earlier.

4.3 Transversal Filter - The Predistorter

It is obvious from the preceding sections that the input signal to the analog filter is digital in nature. This random signal has only two levels of amplitude. A transversal filter, having a shift-register at the input stage is almost a "custom-made" candidate to accept these binary signals for processing in the stages that follow. Hence, a TF comprising a shift-register, a resistive network and a summing device (e.g., OA) could be used to predistort the binary signals before passing them on to the analog filter for bandlimitation.

In Chapter 3 it was discussed that computation of the tap-gain coefficients of the TF is the crux of its design. These coefficients would condition the digital signal in such a way that a single pulse response, seen at the output of the analog filter, would cross the time axis exactly at all sampling points except its own. In case of a random NRZ signal stream this action of the predistorter gives rise to a fully opened eye diagram, i.e , signal with zero ISI, as shown in Fig.4.14.

The tap-gain coefficients of the TF relate directly to the resistive network of the predistorter circuit. The coefficients correspond to the current passing through the resistors that are connected to the shift-register outputs.

Having arbitrarily established the gain of the summing device, it is possible to find the value of each resistor necessary to provide the required current for predistortion. The resistor values may be found in three different ways. In each case the response of the analog filter to a single isolated narrow pulse is investigated and an action is taken outlined as follows.

(a) Hardware method: the response is observed on CRO; the resistor values are found by a "trial and error" technique to yield the desired response.

(b) Hardware/Software method: the response is observed on CRO; the samples are measured with the aid of CRO screen graticule; a software program is used to compute, in an iterative mode, the resistor values needed to minimize the peak distortion as discussed in Chapter 3.

(c) Software Method: the action of the analog filter is simulated on the computer; predistorter tap-gain coefficients (and also resistor values) are computed in a similar manner as outlined in (b).

The above three methods are explained in the following sub-sections.

4.3.1 On Obtaining the Tap-Gains Using Oscilloscope

An arbitrary value for the center tap resistor is chosen. With only this tap connected to the positive input of the summing amplifier the TF is cascaded in front of the bandlimiting filter. The cascaded device is connected to the instruments as shown in Fig.4.8. The test pulse is not distorted at all (except for a delay of one clock period of the shift-register) in reaching the AF for there is no summing action in the summing device, because of the single tap-gain coefficient in the predistorter. Thus, the shape of the analog filter response is the same as that would otherwise result if the TF was bypassed, except for a scaling on the pulse amplitude depending on the gain of the summing amplifier; the relative amplitude of the pulse at all instances are, however, unchanged.

By setting all but one switch to logic 0 on the Word Generator (HP1925A), running in WORD mode, it is possible to generate a repetition of isolated test pulses. Such an isolated test pulse is shown in Fig.4.9. Further, the width of these pulses may be reduced by passing them through a pulse shaper (HP1917A) so that they are narrow enough to consider them approximations to impulses in the ideal situation. The response of the cascaded TF/AF is displayed on the CRO, triggered externally using the same master clock as shown in Fig.4.8. A second trace on the CRO could be

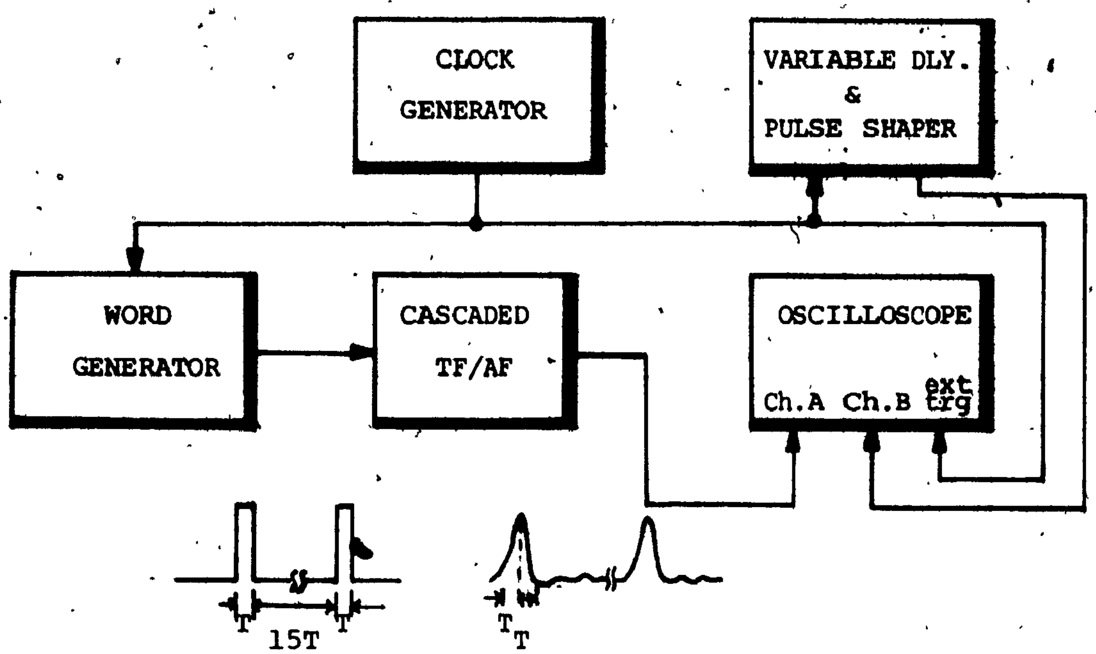
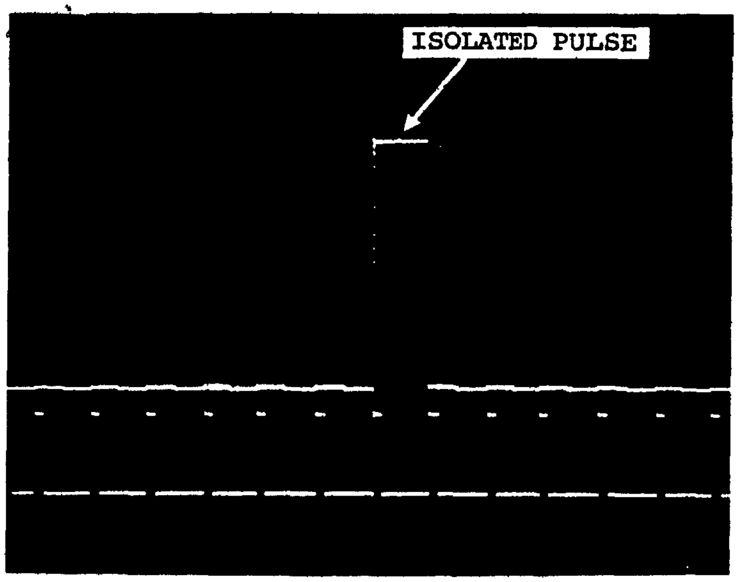


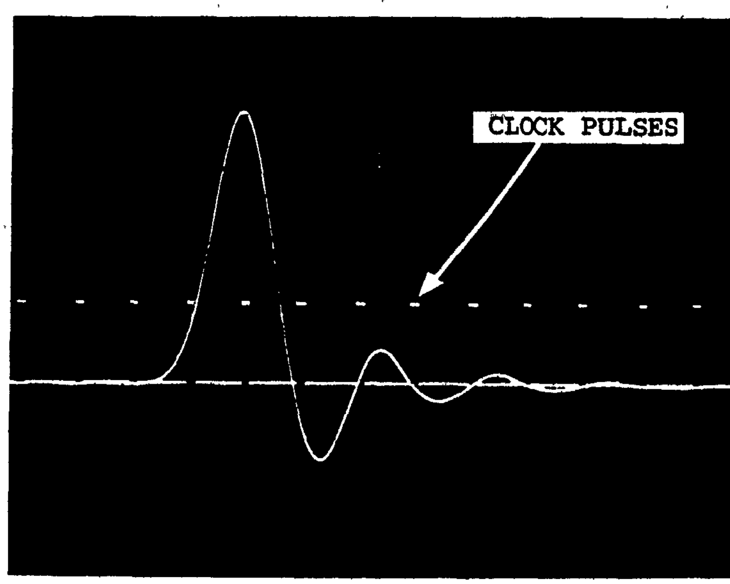
Fig.4.8 Measurement of Pulse Response of AF.

displayed with a train of pulses running at the same clock rate but with controllable width and adjustable delay. This pulse train would be representative of the sampling clock at the receiving end in an actual transmission system. Such a display seen on CRO is shown in Fig.4.10.



(H: 10μs/div , V: > 0.5V/div)

Fig.4.9 Isolated Narrow Pulse as Input to the Cascaded TF/AF.



(H: 2ms/div , V: > 1V/div)

Fig.4.10 Response of the Cascaded TF/AF to the Pulse Shown in Fig.4.9.

One can adjust the delay of the second pulse train so that the leading (or trailing) edge of one clock pulse coincides with the peak of the test pulse. The positions of leading (or trailing) edges of the remaining clock pulses would, then, show the previous and next sampling instants in a receiver environment. It is apparent in the display that pre- and post-cursers of the test pulse do not all cross the time-axis at these sampling instants. As pointed out in Chapter 3 it is imperative that these crossings take place exactly at sampling times if the ISI is to be avoided during a random NRZ signal transmission.

When resistors of various values are connected to the rest of tap positions of the TF, it is interesting to notice how the test pulse response is either pushed up or down depending on the resistor value at each tap as well as the choice of inputs (positive or negative) of the summing device. If fixed resistors in series with trim-pots are used in each tap position, one can see the continuous action of the predistorter when the resistor value is varied by turning the trim-pot screw. This technique, dubbed in the laboratory as "screw-driver optimization," is not the best method, however, in setting the tap-gain coefficients. The reason is that there are an infinite number of tap-gain settings that would yield the desired action of the TF [Lucky et al., 1968]. Yet, this method was described here because of its interesting features that would help understand the concept of TF action.

4.3.2 On Obtaining the Tap-Gains Using Oscilloscope and Computer

This method is essentially the same as the previous one except that the resistor values are computed through an optimization technique rather than a haphazard adjustment of the trim-pots. The instrumentation set-up for this method is the same as in the previous one (see Fig.4.8). The amplitudes of the test pulse at sampling instants are measured using the screen graticule and position knobs of the CRO. The limit to the number of readings taken depends upon the number of times the test pulse crosses the time-axis.

The computer program written for calculating the coefficients assumes that equal number of pulse crossings occur on either side of the pulse peak. This is, of course, a valid assumption for one can assume that the non-existing pulse crossings have zero amplitude. This assumption is necessary if the TF is to have (2N+1) taps, symmetrical about the center tap. In practice, however, there are not more than two non-zero amplitudes ahead of the peak and a varied number after it. As a rule of thumb, it is convenient to consider the number of readings as equal to twice the number of crossings on the time-axis that occur past the peak plus one (counting the peak itself). These values are represented by h_{-i}, $h_{-i+1}, \ldots, h_{-1}, h_0, h_1, \ldots, h_{i-1}, h_i$, i being any integer, as shown in Fig.3.4 of Chapter 3.

The computer program PEAKDIS was written to compute the tap-gain coefficients and, thus, the values of resistors for a specified gain of the summing device, when the sampled amplitude of the test pulse response and the required number of taps in the TF are given as input data. Again, as a rule of thumb, it is convenient to assume that $N = i$, N and i as defined previously. Even if such a rule was not observed (i.e., if $N > i$) the optimization procedure would yield any extra tap-gain coefficients to be zero. However, N must not be less than i, because the optimization technique is based on eqn.(3.11) that defines the peak distortion, for otherwise the residual distortion would not be a minimum.

Reconsider the eqn.(3.12), written here as,

$$h_n = \Sigma_{j=-N}^{N} c_j x_{n-j} \tag{4.1}$$

where h_n represents pulse response amplitude at n^{th} sampling instant, c_j represents tap-gain coefficient at j^{th} tap and x_{n-j} represent input pulse sample to TF at $(n-j)^{th}$ sampling instant. Since ISI is caused by the contribution of all h_i,

$i \neq 0$, all that is necessary here is to force $h_i = 0$, $i \neq 0$, individually by properly choosing tap-gain coefficients c_j. Conversely, optimum values for c_j can be computed now, given the values of h_n and x_{n-j}. Constraining $h_0 = 1$ and $h_i = 0$, $i \neq 0$, eqn.(4.1) can be rewritten in a vector form, to include all n, as

$$\mathbf{1} = \mathbf{C}\mathbf{X} \tag{4.2}$$

where $\mathbf{1}$ is a column vector of length (2N+1) whose elements are all zero except the zeroth, which is unity, $\mathbf{C}$ is also a column vector of length (2N+1) whose j^{th} element is c_j, and $\mathbf{X}$ is a square matrix of order (2N+1), whose ij^{th} element is x_{i-j}. In minimizing peak distortion it was shown in Chapter 3 that the initial peak distortion, D_0, < 1. Thus, here the matrix $\mathbf{X}$ is diagonally dominated and consequently can be shown to possess a non-zero determinant [Lucky et al., 1968]. Again rewriting eqn.(4.2) as

$$0 = \mathbf{1} - \mathbf{C}\mathbf{X} \tag{4.3}$$

one would find a solution for $\mathbf{C}$ by which eqn.(4.3) holds true. There is a very good possibility that one may wish to use a fewer number of taps in the TF than the number of non-zero amplitude samples of the response. In such a case eqn.(4.3) would become an overdetermined system of n linear equations in (2N+1) unknowns. It has also been found experimentally that more the number of taps there are, the more number of tails result in the final response. Therefore, it was found easy and efficient in this work to compute (2N+1) variables in C by treating eqn.(4.3) as an overdetermined system of n linear equations, where n > (2N+1). However, in computation it is a good practice to equate the right hand side of eqn.(4.1) to an error function e(x) and then compute C in an iterative mode such that e(x) is minimized to a tolerable value. Thus, an algorithm written by Barrodale and Roberts [1974] for such a system of equations was adopted to compute the optimum tap-gain coefficients that would force all ISI contributors to zero amplitude when the TF and analog filter are connected in tandem. The tolerance e(x) was set at 10^{-10} in running PEAKDIS. The results obtained are shown in Table 4.1. When the computed resistor values are used in the TF, the improvement obtained in a pulse response as well as in the eye diagram of a random NRZ signal are shown in Figs.4.11 through 4.14. The degradation in the peak eye opening, here, caused by ISI is only 0.43 dB. The improvement in the eye opening is, therefore, 4.14 dB when Fig.4.6 is compared with Fig.4.14.

Table 4.1 Results of Program PEAKDIS

INPUT DATA

NO. TAPS IN THE EQUALIZER IS (2N+1), WHERE N = 13
NO. PULSE SAMPLES REQUIRED FOR COMPUTATION = 53
TOLERANCE IN COMPUTATION = [illegible]

INPUT PULSE SAMPLES GIVEN AS.

X(-13) =	0.000000
X(-12) =	0.000000
X(-11) =	0.000000
X(-10) =	0.000000
X(-9) =	0.000000
X(-8) =	0.000000
X(-7) =	0.000000
X(-6) =	0.000000
X(-5) =	0.000000
X(-4) =	0.002400
X(-3) =	0.000000
X(-2) =	-.021060
X(-1) =	.171075
X(0) =	.001596
X(1) =	-.017010
X(2) =	-.049611
X(3) =	.017947
X(4) =	-.030017
X(5) =	.019596
X(6) =	-.004240
X(7) =	.001699
X(8) =	-.010667
X(9) =	.004207
X(10) =	.004413
X(11) =	.001124
X(12) =	-.000794
X(13) =	.001059

ALL OTHER SAMPLES ARE ASSUMED TO BE ZERO.

SAMPLES AFTER NORMALIZATION BY THE CENTER SAMPLE, X(0).

X(-26) =	0.000000
X(-25) =	0.000000
X(-24) =	0.000000
X(-23) =	0.000000
X(-22) =	0.000000
X(-21) =	0.000000
X(-20) =	0.000000
X(-19) =	0.000000
X(-18) =	0.000000
X(-17) =	0.000000
X(-16) =	0.000000
X(-15) =	0.000000
X(-14) =	0.000000
X(-13) =	0.000000
X(-12) =	0.000000
X(-11) =	0.000000
X(-10) =	0.000000
X(-9) =	0.000000
X(-8) =	0.000000
X(-7) =	0.000000
X(-6) =	0.000000
X(-5) =	0.000000
X(-4) =	0.000000
X(-3) =	[illegible]
X(-2) =	-.021519
X(-1) =	.102693
X(0) =	1.000000
X(1) =	-.011963
X(2) =	-.031060
X(3) =	[illegible]
X(4) =	-.006660
X(5) =	.021979
X(6) =	-.007065
X(7) =	.004160
X(8) =	-.000740
X(9) =	[illegible]
X(10) =	.007195
X(11) =	.000139
X(12) =	[illegible]
X(13) =	.001082
X(14) =	0.000000
X(15) =	0.000000
X(16) =	0.000000
X(17) =	0.000000
X(18) =	0.000000
X(19) =	0.000000
X(20) =	0.000000
X(21) =	0.000000
X(22) =	0.000000
X(23) =	0.000000
X(24) =	0.000000
X(25) =	0.000000
X(26) =	0.000000

RESULTS

MINIMUM SUM OF ABSOLUTE VALUES OF THE RESIDUALS = [illegible]
THE RANK OF THE MATRIX 'A' = 27
EXIT CODE OF THE SUBROUTINE = 0
NUMBER OF ITERATIONS REQUIRED BY THE SIMPLEX METHOD = 27

UNIQUE OPTIMAL SOLUTION IS FOUND.

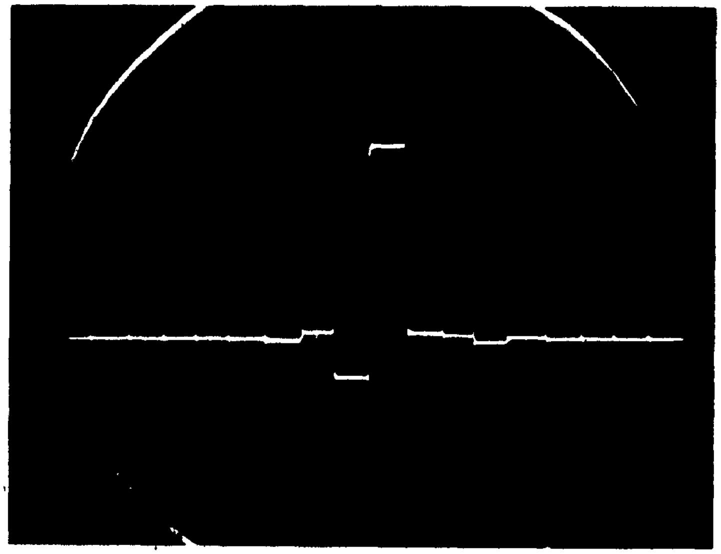
SOLUTION TO THE PROBLEM.

ARRAY C CONTAINING THE TAP-GAIN COEFFICIENTS

C(-13) =	[illegible]
C(-12) =	[illegible]
C(-11) =	[illegible]
C(-10) =	[illegible]
C(-9) =	[illegible]
C(-8) =	[illegible]
C(-7) =	[illegible]
C(-6) =	[illegible]
C(-5) =	[illegible]
C(-4) =	[illegible]
C(-3) =	[illegible]
C(-2) =	[illegible]
C(-1) =	[illegible]
C(0) =	[illegible]
C(1) =	[illegible]
C(2) =	[illegible]
C(3) =	[illegible]
C(4) =	[illegible]
C(5) =	[illegible]
C(6) =	[illegible]
C(7) =	[illegible]
C(8) =	[illegible]
C(9) =	[illegible]
C(10) =	[illegible]
C(11) =	[illegible]
C(12) =	[illegible]
C(13) =	[illegible]
R(1) =	0.
R(2) =	0.
R(3) =	0.
R(4) =	0.
R(5) =	0.
R(6) =	0.
R(7) =	0.
R(8) =	0.
R(9) =	0.
R(10) =	0.
R(11) =	0.
R(12) =	0.
R(13) =	0.
R(14) =	0.
R(15) =	0.
R(16) =	0.
R(17) =	0.
R(18) =	0.
R(19) =	0.
R(20) =	0.
R(21) =	0.
R(22) =	0.
R(23) =	0.
R(24) =	0.
R(25) =	0.
R(26) =	0.
R(27) =	0.

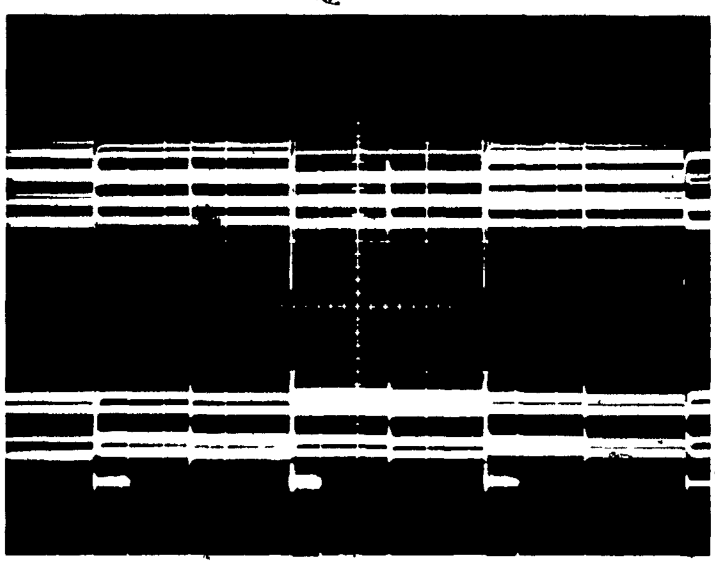
COEFFICIENTS ARE NORMALIZED WITH RESPECT TO THE CENTER COEFFICIENT, C(0).

NORMALIZED COEFFICIENTS		CORRESPONDING RESISTOR VALUE		
C(-13) =	[illegible]	R(-13) =	[illegible]	KILO-OHMS
C(-12) =	[illegible]	R(-12) =	[illegible]	KILO-OHMS
C(-11) =	[illegible]	R(-11) =	[illegible]	KILO-OHMS
C(-10) =	[illegible]	R(-10) =	[illegible]	KILO-OHMS
C(-9) =	[illegible]	R(-9) =	[illegible]	KILO-OHMS
C(-8) =	[illegible]	R(-8) =	[illegible]	KILO-OHMS
C(-7) =	[illegible]	R(-7) =	[illegible]	KILO-OHMS
C(-6) =	[illegible]	R(-6) =	[illegible]	KILO-OHMS
C(-5) =	[illegible]	R(-5) =	[illegible]	KILO-OHMS
C(-4) =	[illegible]	R(-4) =	[illegible]	KILO-OHMS
C(-3) =	[illegible]	R(-3) =	[illegible]	KILO-OHMS
C(-2) =	[illegible]	R(-2) =	[illegible]	KILO-OHMS
C(-1) =	[illegible]	R(-1) =	[illegible]	KILO-OHMS
C(0) =	.100000000E+01	R(0) =	[illegible]	KILO-OHMS
C(1) =	[illegible]	R(1) =	[illegible]	KILO-OHMS
C(2) =	[illegible]	R(2) =	[illegible]	KILO-OHMS
C(3) =	[illegible]	R(3) =	[illegible]	KILO-OHMS
C(4) =	[illegible]	R(4) =	[illegible]	KILO-OHMS
C(5) =	[illegible]	R(5) =	[illegible]	KILO-OHMS
C(6) =	[illegible]	R(6) =	[illegible]	KILO-OHMS
C(7) =	[illegible]	R(7) =	[illegible]	KILO-OHMS
C(8) =	[illegible]	R(8) =	[illegible]	KILO-OHMS
C(9) =	[illegible]	R(9) =	[illegible]	KILO-OHMS
C(10) =	[illegible]	R(10) =	[illegible]	KILO-OHMS
C(11) =	[illegible]	R(11) =	[illegible]	KILO-OHMS
C(12) =	[illegible]	R(12) =	[illegible]	KILO-OHMS
C(13) =	[illegible]	R(13) =	[illegible]	KILO-OHMS



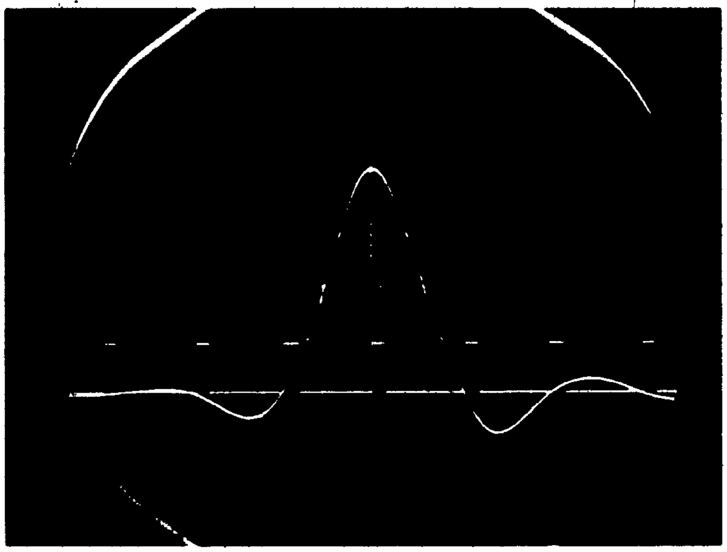
(H: > 20μs/div , V: >1V/div)

Fig.4.11 Predistorted Pulse, i.e., the Response of TF to a Single Isolated Pulse.



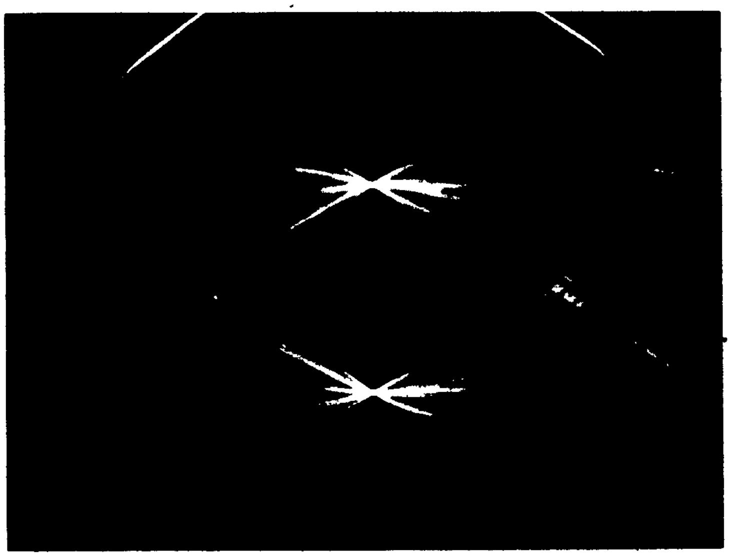
H: > 0.02ms/div , V: > 0.5V/div & 1V/div

Fig.4.12 Predistorted PRBS (Upper Trace) and 32 kHz Clock Pulses (Lower Trace).



(H: >20μs/div , V: >1V/div)

Fig.4.13 The Response of AF to the Predistorted Pulse Shown in Fig.4.11.



(H: 5μs/div , V: >1V/div)

Fig.4.14 The Eye Diagram of Predistorted PRBS (i.e., the Response of Cascaded TF/AF to PRBS)

4.3.3 On Obtaining the Tap-Gains Using Computer

The method described here does not call for the usage of CRO at all. The test pulse that was displayed on CRO in the two previous methods can, instead, be simulated on the computer if the amplitude and phase characteristics of the

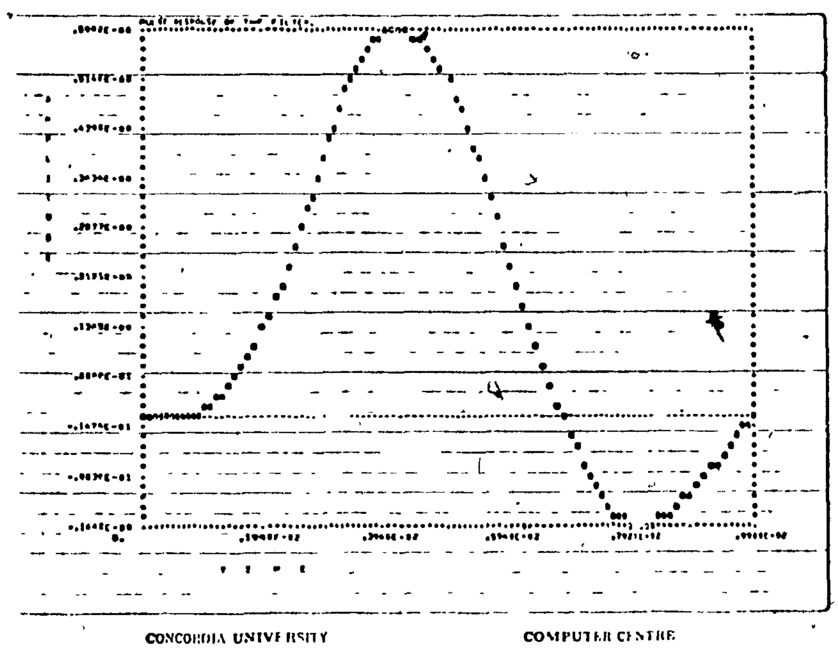


Fig.4.15 Simulated Test Pulse Response.

analog filter are known. The amplitude (absolute, not in decibels) and phase (in degrees) characteristics of analog filter as shown in Fig.4.3 were used in the program CURVFIT to compute the coefficients of a 13-degree polynomial equation for each curve. These coefficients were then supplied to the program PULSE that simulated the test pulse response identical (within computational error) to what was obtained through the hardware circuit. The simulated pulse response is shown in Fig.4.15. The plotting subroutine truncated the pulse at the second lobe after the peak because this subroutine accepts only the first 100 points of the abscissa. The samples of the test pulse are also

Table 4.2 Samples of Simulated Test Pulse Response

Sample No.	Sample Value	Remarks
1	0.0123	Left of Peak
2	0.3106	-do-
3	0.5902	The Peak
4	0.2646	Right of Peak
5	-0.1329	-do-
6	-0.0941	-do-
7	0.0659	-do-
8	0.0260	-do-
9	-0.0561	-do-
10	-0.0199	-do-
11	0.0319	-do-
12	0.0111	-do-
13	-0.0178	-do-
14	-0.0051	-do-
15	0.0107	-do-
16	0.0025	-do-

printed as results in this program, as shown in Table 4.2. These samples are then used in the program PEAKDIS as in the second method to compute the optimum tap-gain coefficients and, hence, resistor values for the specified gain of the summing amplifier.

In order to accept the validity of simulation, all computer programs were tested with certain known parameters to yield the expected results as described in the next two paragraphs. As explained in the Appendix C, the signal processing in SIMULA was performed on sections of PRBS, one at a time, instead of in complete sequence. However, as the number of steps required for processing increases the program steps become extremely complicated owing to the necessity for simultaneous processing of a number of sections. In this application, as the simulation is done only at the transmit end of a baseband transmission system, complications were minor.

SIMULA was, first, tested to see the impact of predistortion. If all but one tap-gain coefficient of the TF are set to zero the signal at TF output cannot be distorted, except for a scaling owing to the non-zero tap-gain and the gain of the summing device. In simulation, the scaling effect can be totally eliminated by setting the single tap-gain coefficient to unity. (The summing action in the simulation is simply an arithmetic addition; thus, summing gain is also unity.) This is equivalent to bypassing the TF. Hence, the program was tested with and without TF. In the former case, five* taps were used, with the center

---

* The program SIMULA has a limitation with regard to the number of taps that could be used in the TF, i.e., 5 $\leq$ no. taps $\leq$ 27 and it must be an odd integer.

tap-gain set to unity and others set to zero. In the second case, all the simulation steps associated with predistortion process** were deleted. The results so obtained were identical, thereby, showing the accuracy of predistortion steps in the program.

The next step was to test the simulation with a known transmit filter. A raised cosine filter, modified with an x/sin(x) function, is well suited for this purpose. The eye diagram at the output of such a Nyquist filter would have zero ISI [Bennett and Davey, Ch.3, 1965]. Furthermore, when α, the excess Nyquist bandwidth, equals unity the jitter at the time-axis crossings would also be zero; but, the jitter increases as α decreases $(0 < \alpha \leq 1)$ [Feher, Ch.3, 1981]. Eye diagrams for α = 1.0, 0.4 and 0.3 are shown in Fig.4.16. They confirm the Nyquist criteria for digital transmission. These two tests prove the validity of the simulation program SIMULA.

** These include subroutines PREDIST & ADTAIL2 and intermediate storing of data in TAPE3 (see Appendix C). Only two sections of the signal are simultaneously processed in each loop.

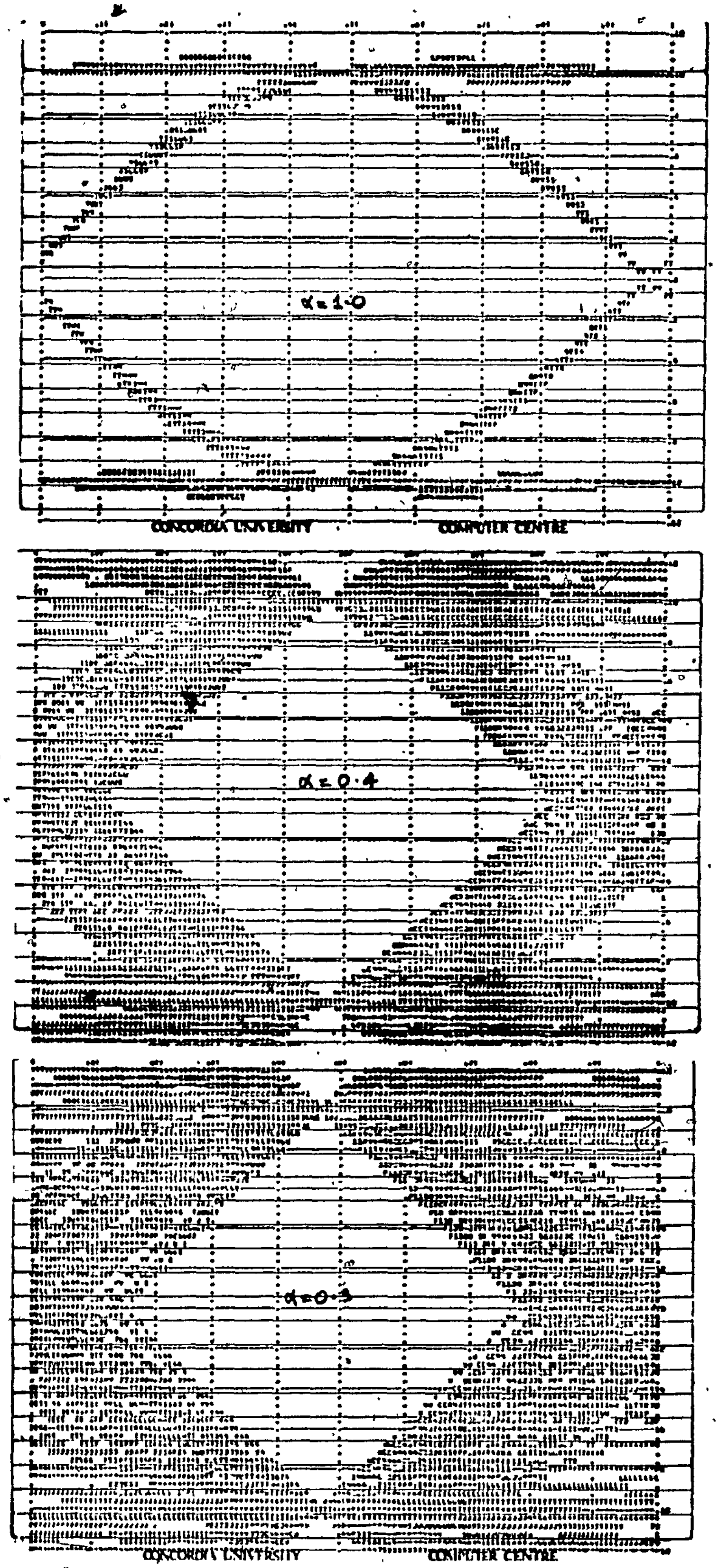
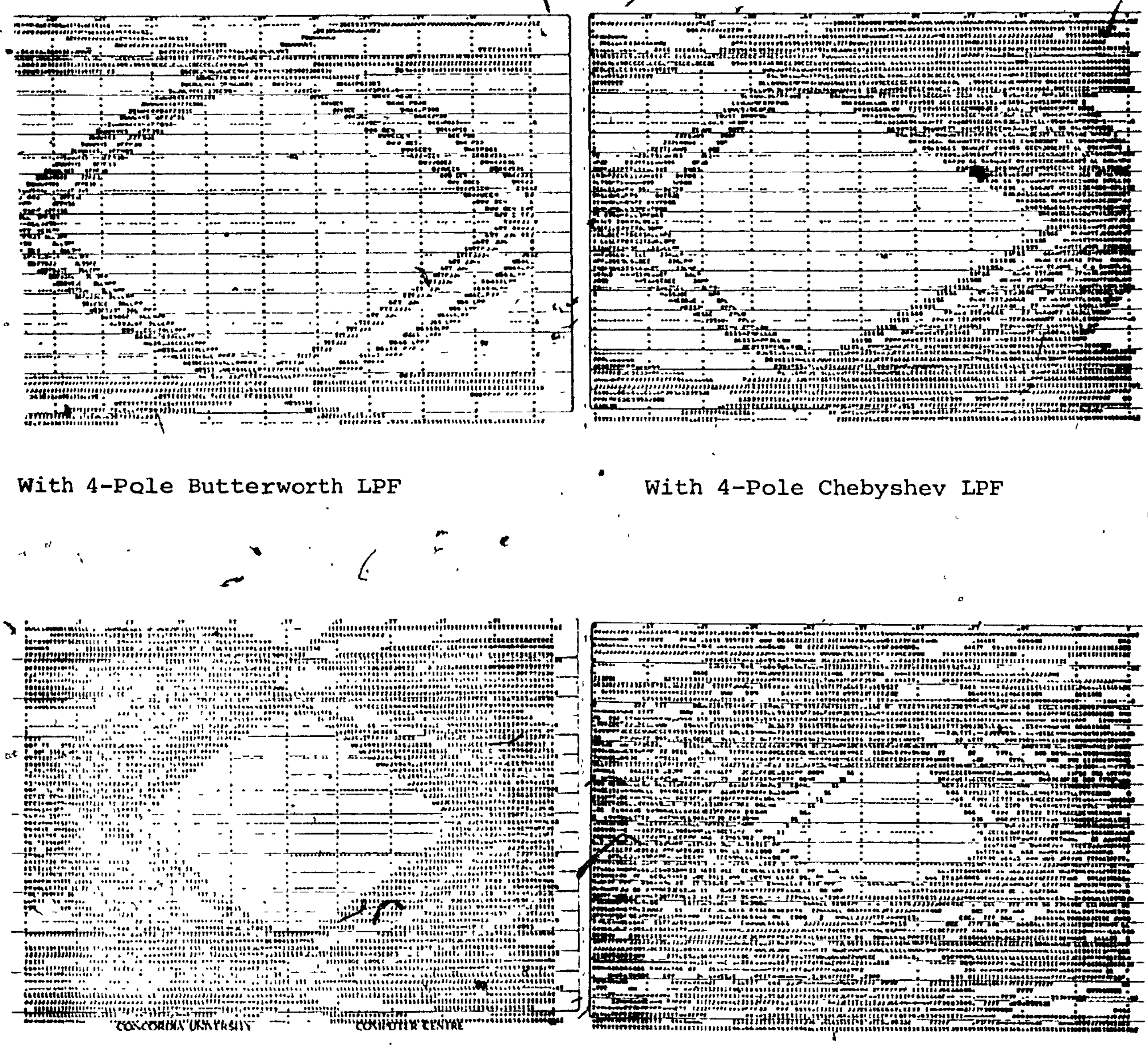


Fig.4.16 Simulated Eye Diagrams of Raised Cosine Filters with α = 1.0, 0.4 and 0.3 (α is the Excess Nyquist Bandwidth)

Fig.4.17 shows the eye diagrams (no predistortion) obtained with different types of conventional analog filters



With 4-Pole Butterworth LPF

With 4-Pole Chebyshev LPF

With 8-Pole Butterworth LPF

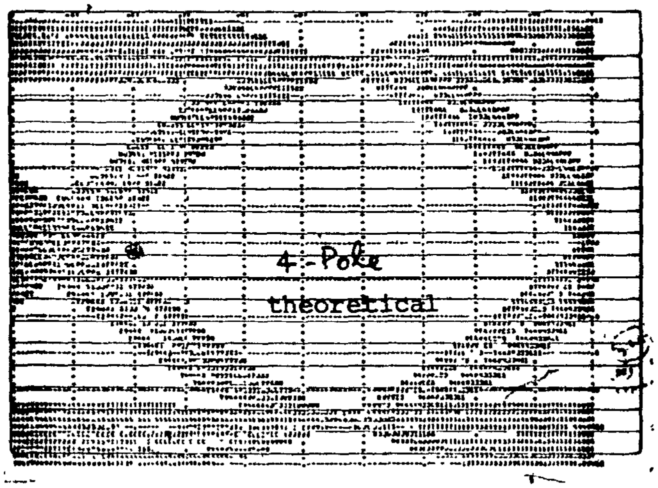
With 8-Pole Chebyshev LPF

Fig.17 Simulated Eye Diagrams of Conventional Theoretical Filters (4- and 8-Pole Butterworth, Chebyshev Low Pass Filters).

when theoretical parameters were used. Butterworth and Chebyshev filters of order n=4 and 8 were used (with ripple = 0.5 dB in the Chebyshev case). The four eyes indicate heavy ISI as n is increased and are as expected for these filters were not equalized. Two of these filters were used in PULSE to obtain samples of its pulse response which were, then, used in PEAKDIS to compute optimum tap-gain coefficients required for the TF. These coefficients were used for predistortion when SIMULA was executed again using each filter. The resulting eye diagrams, as shown in Fig.4.18, show remarkable improvement.

Lastly, a practical filter was used in the simulation. The 8th-order Butterworth filter, whose amplitude and phase characteristics are shown in Fig.4.3, was used here. These characteristics were supplied to CURVFIT to obtain the 13-degree polynomials which were, then, used in SIMULA to print the eye diagram without predistortion as in the theoretical case. PULSE and PEAKDIS were re-executed to compute the tap-gain coefficients as explained earlier. SIMULA was re-run using these coefficients for predistortion in view of obtaining an improved eye. The unequalized and equalized eyes of the practical filter are shown in Fig.4.19.

If Figs.4.18 and 4.19 are compared a difference in the degree of improvement in the equalized eyes may be observed.



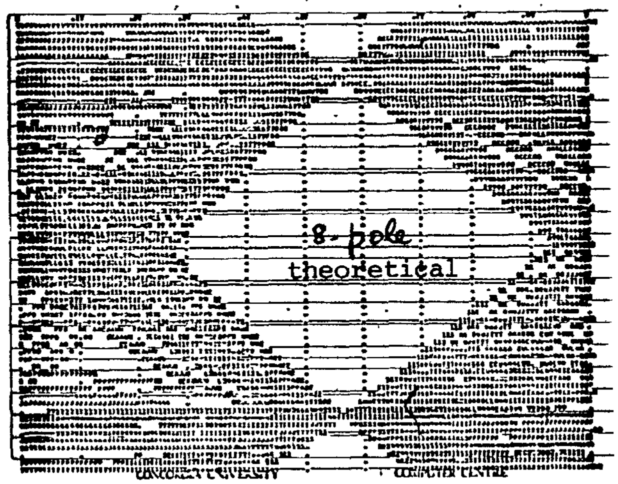


Fig.4.18 Simulated Improved Eye Diagrams of 4- and 8-Pole Butterworth LPFs.

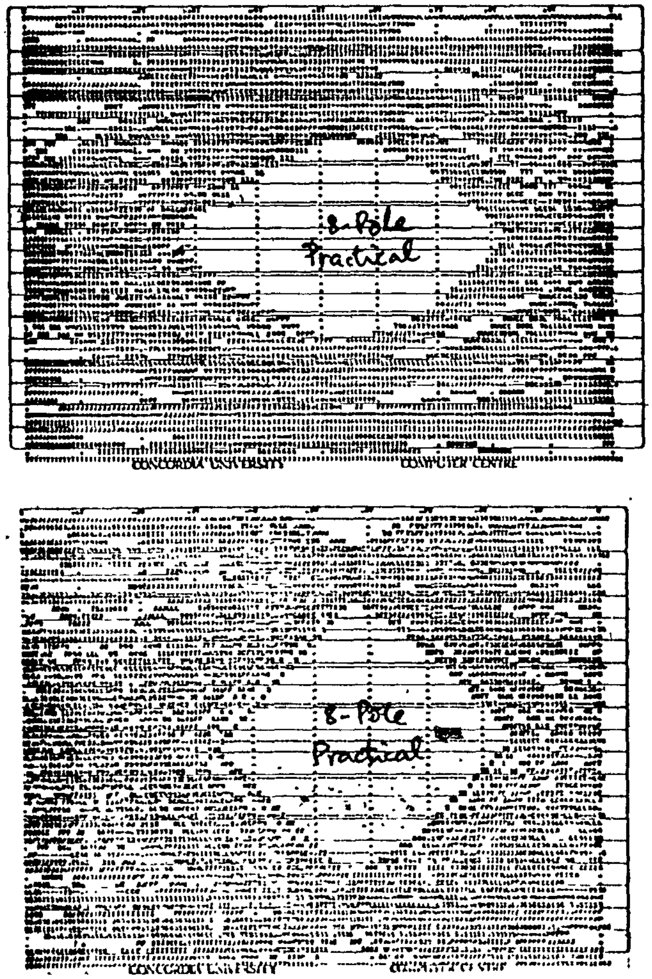


Fig.4.19 Simulated Eye Diagrams of a Practical 8-Pole Butterworth LPF.

The theoretical filter eye has been improved better than the practical. The reason for this discrepancy lies in CURVFIT. It was found that the set of polynomials fitted by this program to a given curve was not accurate enough. This is observed in the plots shown in Figs.4.20 and 4.21 for both amplitude and phase characteristics, when compared with actual characteristics (of Fig.4.3) shown here by dotted

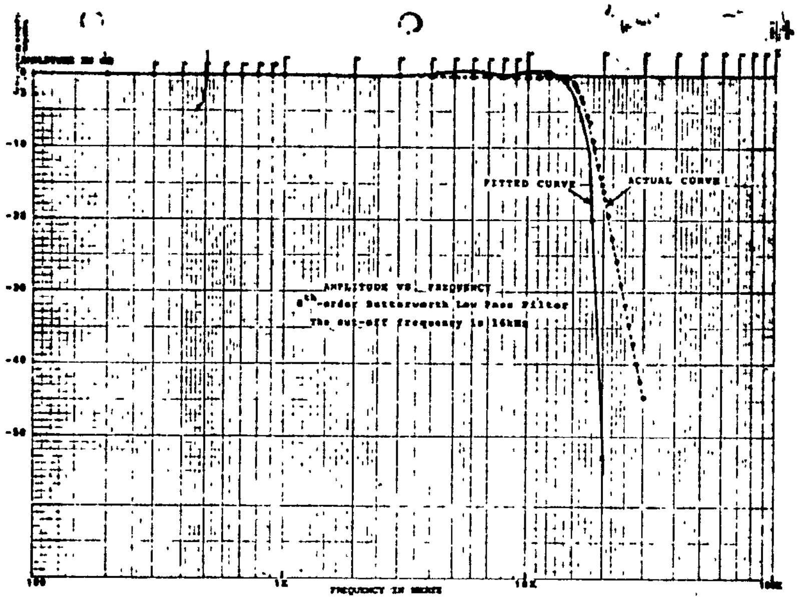


Fig.4.20 The Actual and Fitted Curves of Gain vs. Frequency Characteristic of the Practical 8-Pole Butterowth LPF.

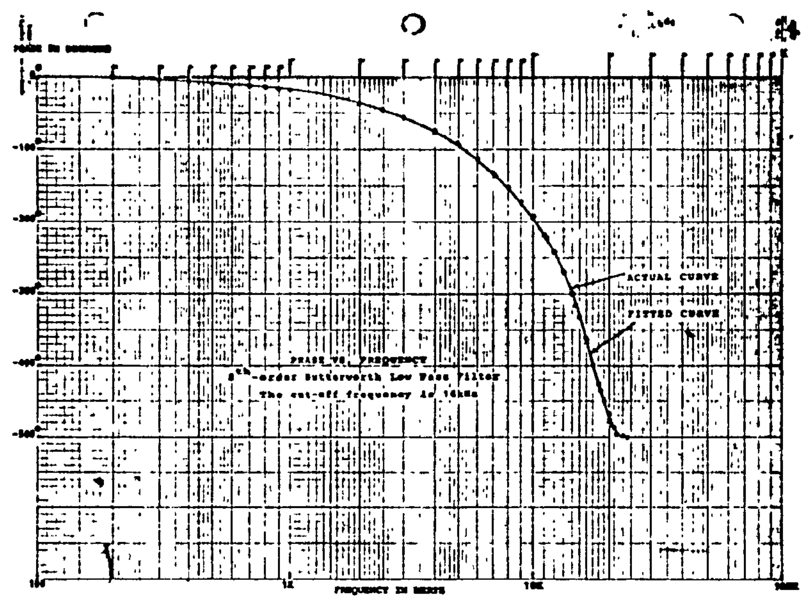


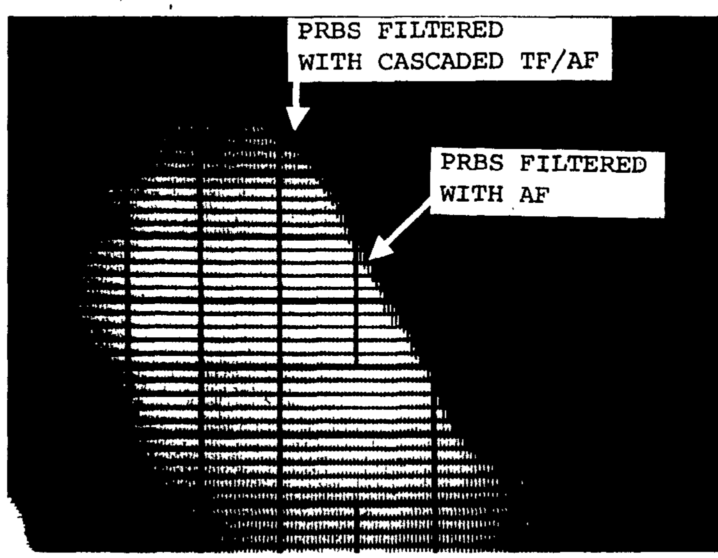
Fig.4.21 The Actual and Fitted Curves of Phase vs. Frequency Characteristic of the Practical 8-Pole Butterworth LPF.

lines. Therefore, computational errors in each step of the total simulation, i.e., all programs mentioned in this section, contribute to some residual ISI in the equalized eye. It is anticipated that better results could be achieved if an efficient curve fitting program (e.g., using spline techniques) would be available.

In spite of the few drawbacks mentioned, the simulation results are encouraging. The programs, listed in Appendix C, are powerful tools for making a decision on the required baseband transmit filter before actually constructing the hardware unit. To sum up, if resistor values computed through the simulation technique, mentioned in this section, are used in the TF of the hardware unit, then tuning of resistors may be necessary to completely eliminate the ISI. A CRO is required for this purpose.

4.4 Performance Evaluation of the Cascaded Filter

In the previous sections it was shown how the analog filter response could be improved by predistorting the digital signal through the use of a TF cascaded in front. The improvements both in a single test pulse case and during a random binary NRZ signal case were shown step by step. The basic reason for using the analog filter, in the first place, was to bandlimit the digital signal before transmission for obvious reasons mentioned in Chapter 1. Because of the time-domain degradation caused by this bandlimiting filter, a TF was used to improve the quality of the signal.



(H: 5kHz/div , V: 10dB/div)

Fig.4.22 Power Spectra of Filtered Digital Signal (a) without and (b) with TF Cascaded in Front of Analog Filter. In-Band Power is Increased by the TF Action.

The question that arises now is whether this TF has any impact on the power spectrum of the signal. If the answer is 'NO' one could live with this design. However, if it is 'YES', a second question arises. Has the TF so designed caused damage on the spectrum? The answer to this question, fortunately, is also 'NO'; in fact, the TF has improved the signal spectrum too by increasing the in-band power, as shown in Fig.4.22. This increase of in-band power is due to the fact that the TF acts as a whitening filter on the random signal, with a transfer function of $x/\sin(x)$

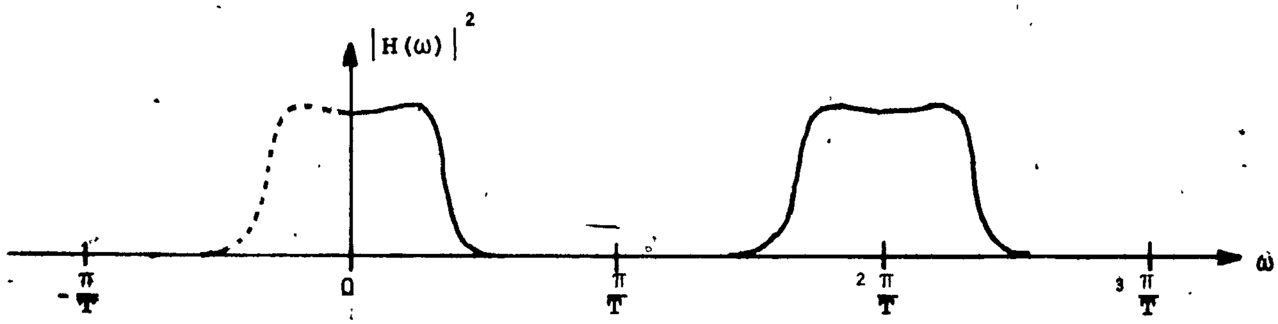


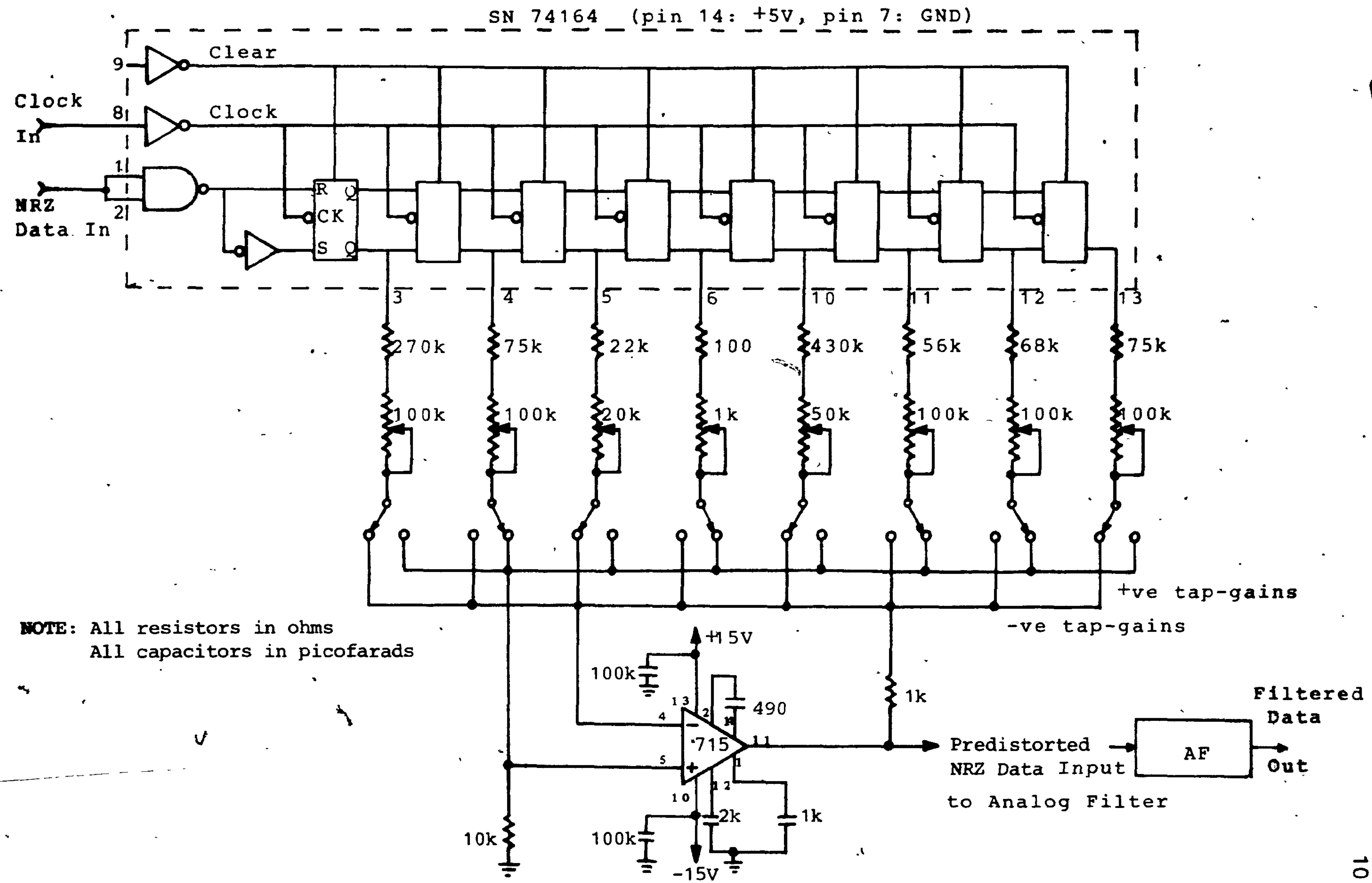
Fig.4.23 Transfer Function of a Transversal Filter.

characteristic, as shown in Fig.4.23. This transfer function is obtained by considering the response function given by eqn.(3.4), rewritten here as

$$E(\omega) = \Sigma_{i=-N}^{N} c_i e^{-j\omega iT} \qquad (4.4)$$

Using computed values of c_i, the results of Section 4.3.2, the response function is plotted in Fig.4.23 for a range of values of ω. The complete schematic diagram of the cascaded transversal/active analog low pass filter is shown in Fig.4.24.

The eye diagram and power spectrum of a random NRZ signal are two important aspects of judging the action of a transmit filter, especially in a laboratory environment. They also provide quick first-hand information on the



NOTE: All resistors in ohms
All capacitors in picofarads

Fig.4.24 The Cascaded Digital/Analog Low Pass Filter.

behaviour of the device in a transmission system. In a digital transmission system these tests, however, do not provide a complete assessment of such a device. One major reason would be that it is not always apparent through these tests whether or not the actual values of discrete components, such as resistors and capacitors, have drifted away from their initial values as the ambient temperature varies over time. The most common test that is performed, therefore, to fulfill this requirement is the measurement of bit error rate (BER), or the probability of error (P(e)), in the system. The BER is measured, experimentally, as defined by the equation,

$$BER = N_e/N_t = N_e/f_b t_0 \qquad (4.5)$$

where N_e is number of bit errors in a time interval t_0, N_t is the total number of bits transmitted in time t_0, f_b is the bit rate of the data source and t_0 is the measuring time interval.

For a random, stationary error generation process and sufficiently long measurement interval t_0, the measured BER gives an estimate of the true P(e). The BER measurement is executed with a pseudo-random test signal sequence which is transmitted through the channel. The receiver computes P(e) by comparing the received bits with a stored replica of the transmitted bit pattern. An instrumentation set-up to measure BER in an additive white gaussian noise (AWGN)

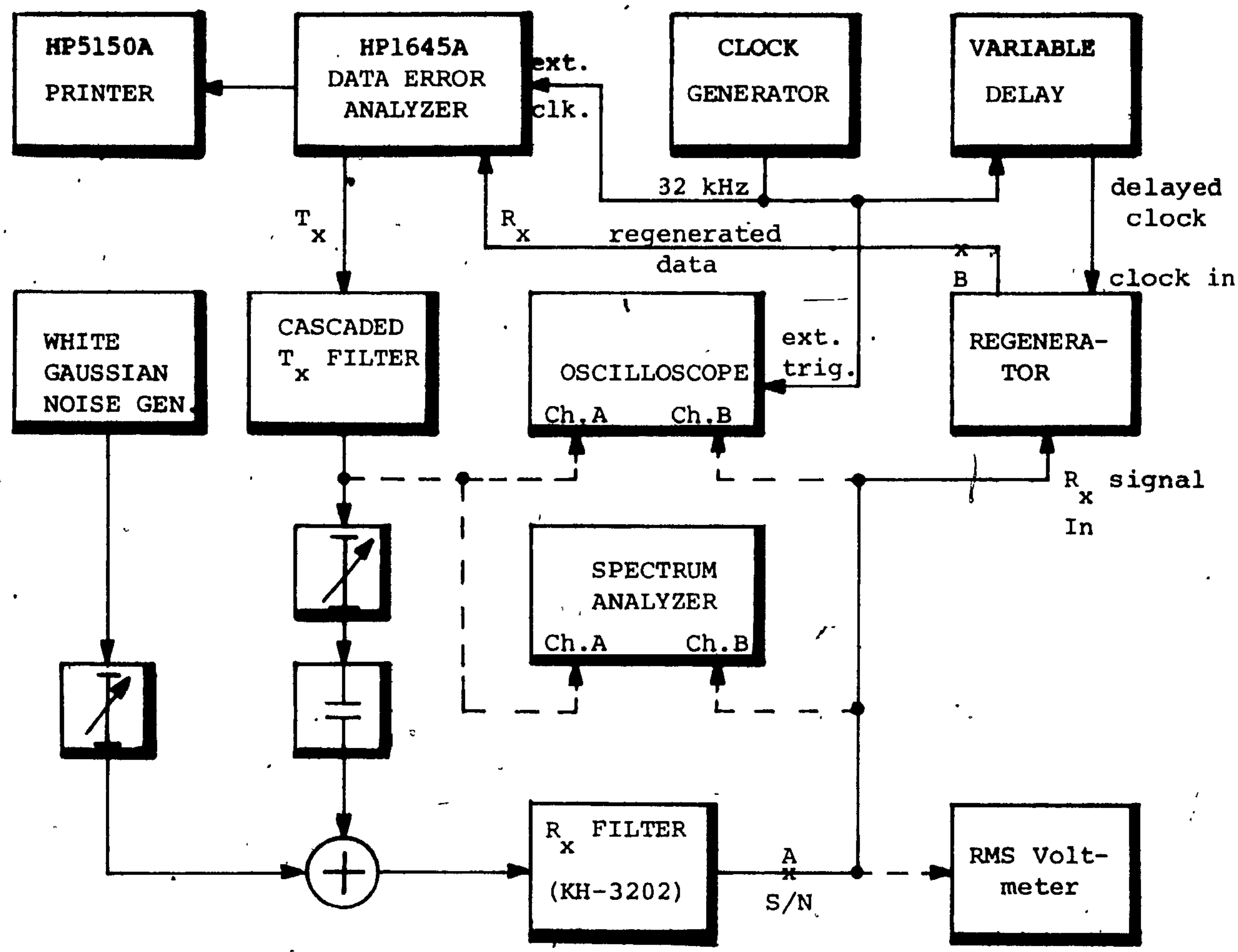


Fig.4.25 Block Diagram of the Set-Up for Measuring the P(e).

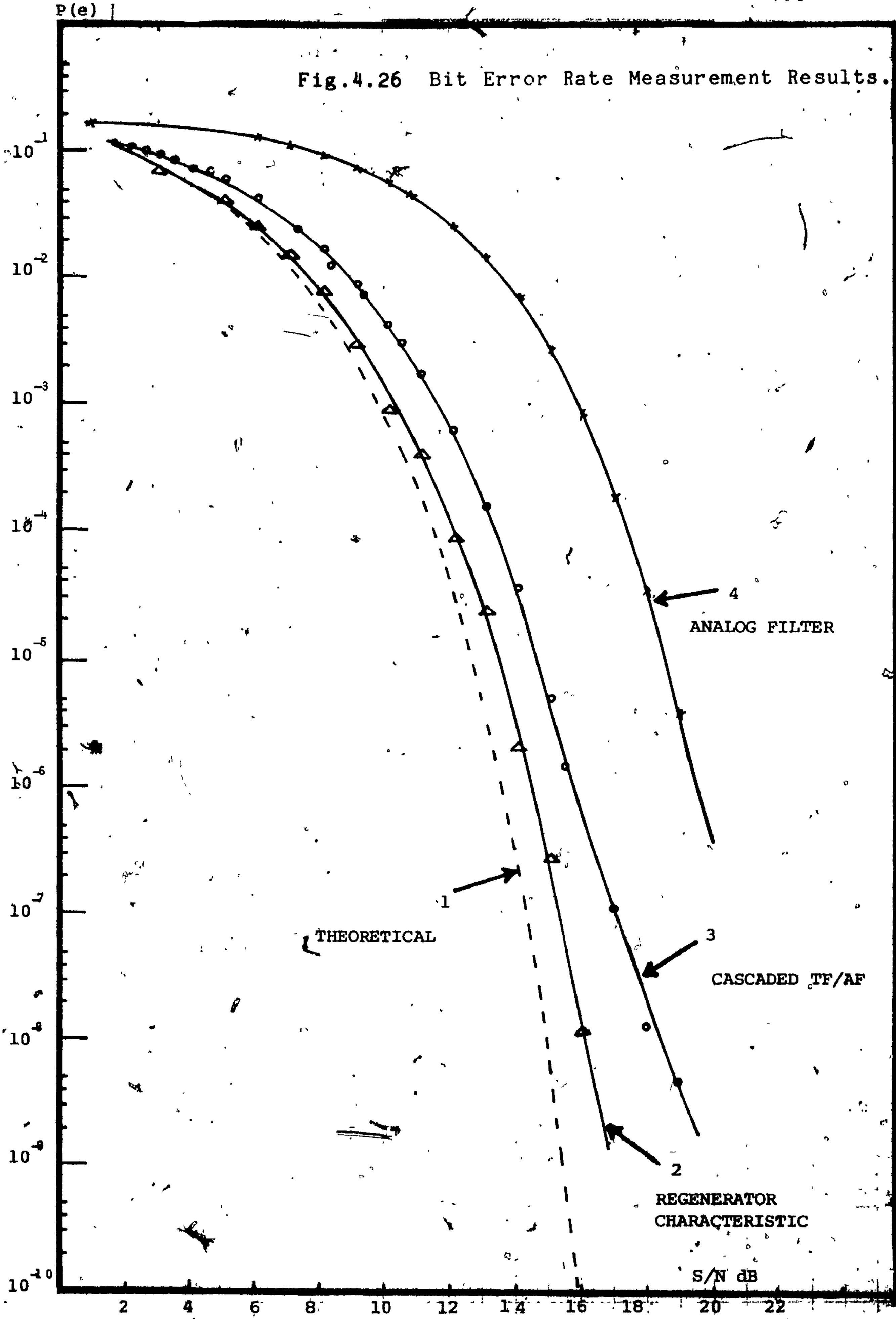
channel (simulated) is shown in Fig.4.25.

In this test set-up the Data Error Analyzer (HP1645A) was clocked externally with a 32 kHz repetitive square wave signal to generate a 32 kb/s pseudo-random binary signal. This signal was filtered using the cascaded digital/analog filter and passed through a large capacitor to remove any DC voltage present in the signal. White gaussian noise was added to the signal before it was passed through the receive low pass filter having a cut-off frequency of 20 kHz. The receive filter used was a 4^{th}-order Butterworth type (Krohn-Hite model 3202) bench unit. The output signal of this receive filter was fed into the regenerator (threshold device) which was clocked by the same master clock driving the data source (HP1645A) but with a variable delay. This delay is essential to adjust the clock position so that the regenerator could sample the received signal at its maximum eye opening.

The regenerated data was fed back into HP1645A to measure the bit error rate as recorded on HP5150A Thermal Printer, for different settings of the attenuators along the transmission path. The attenuators are used to control the signal power and the noise power, thus, giving measurements of P(e) at various signal-to-noise ratios (S/N). The signal and noise power were measured with a true rms voltmeter at point A as shown in Fig.4.25. The transmit and receive signals in both time- and frequency-domains were displayed on CRO and Spectrum Analyzer, respectively, at points along the transmission path as marked with dashed lines in Fig.4.25. The results of this test were plotted as shown in Fig.4.26. The dashed line is the theoretical curve obtained from the equation for probability of error, P(e), given as

$$P(e) = 0.5 \operatorname{erfc} \frac{\hat{S}/N}{\sqrt{2}} \tag{4.33}$$

where N is the rms value of the noise and $\hat{S}$ is the peak value of the signal, which also corresponds to S, the rms value of the unfiltered PRBS signal in the case when the signal suffers no bandlimiting efffects in the transmission process [Brind'Amour and Feher, 1980]. However, it is important to note that the regenerator that is used to restore the discrete form of the signal shape has its own imperfections. Therefore, the regenerator was also evaluated by measuring the BER of infinite bandwidth PRBS (i.e., bypassing the transmit filter) but using the same receive filter as used in the other measurements. These measurement results are depicted by the solid curve marked 2. It can be noticed in the figure that this measured characteristic follows the theoretical curve with a maximum deviation of 1.0 dB at a measured BER value of 1×10^{-8}. This curve is good enough in a practical situation such as here. The two curves marked 3 and 4, respectively, depict the measurement results when the analog transmit filter was used with and without the predistorter (transversal filter) cascaded to it. An improvement of 4.5 dB at BER = 1×10^{-4} and of 4.0 dB at BER = 1×10^{-6} in S/N can be achieved when the predistorter is used cascaded to the analog transmit filter.

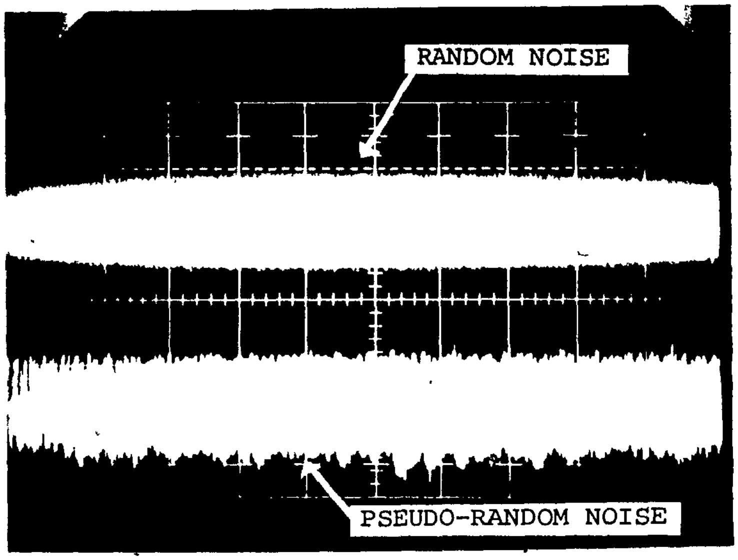


P(e)
Fig.4.26 Bit Error Rate Measurement Results.
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
4
ANALOG FILTER
1
THEORETICAL
3
CASCADED TF/AF
2
REGENERATOR
CHARACTERISTIC
S/N dB
2
4
6
8
10
12
14
16
18
20
22

4.5 Performance Evaluation of the Pseudo-Random Noise Generator

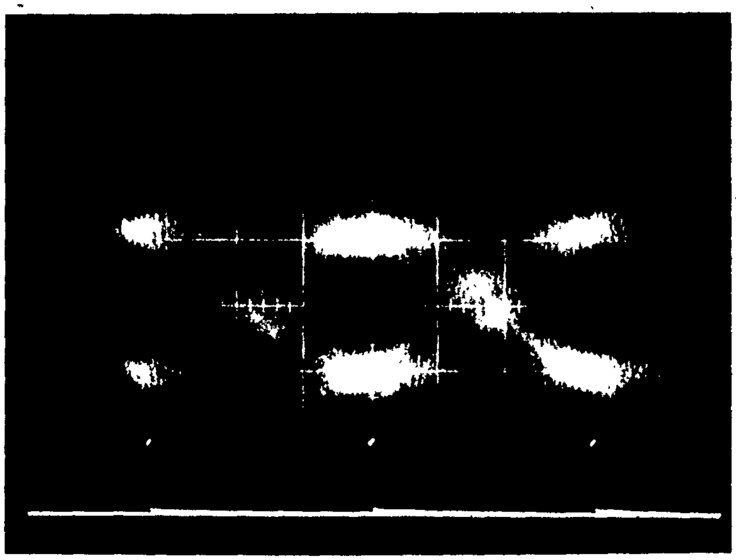
In Chapter 2 a description was given on the design of a laboratory-built unit of a noise generator that would exhibit gaussian amplitude characteristics if certain design criteria are followed, i.e., the PRBS stream must be of sufficiently long maximum-length sequence and the simple RC low pass filter cut-off frequency must be one-twentieth of the PRBS bit rate. In this section some results are shown with regard to the performance of this simple and low-cost noise generator in a baseband transmission system. The performance is compared with that of a conventional source of white gaussian noise.

Fig.4.27 shows a photograph of a CRO display depicting the random noise (upper trace) and pseudo-random noise (lower trace) generated by a conventional (GR1383 Random Noise Generator) and the prototype units, respectively. The PRBS in the prototype unit was generated at 500 kb/s and was filtered with an RC LPF of f_c = 25 kHz. Fig.4.28 shows the eye diagram of the transmitted signal (bandlimited) when random noise was added prior to the receive filter. The ISI in the signal is due to the transmit filter because predistortion was not used when this photograph was taken. Fig.4.29 is a similar display but the noise added was pseudo-random in nature, as supplied by the prototype unit. These displays show the difference between the two types of



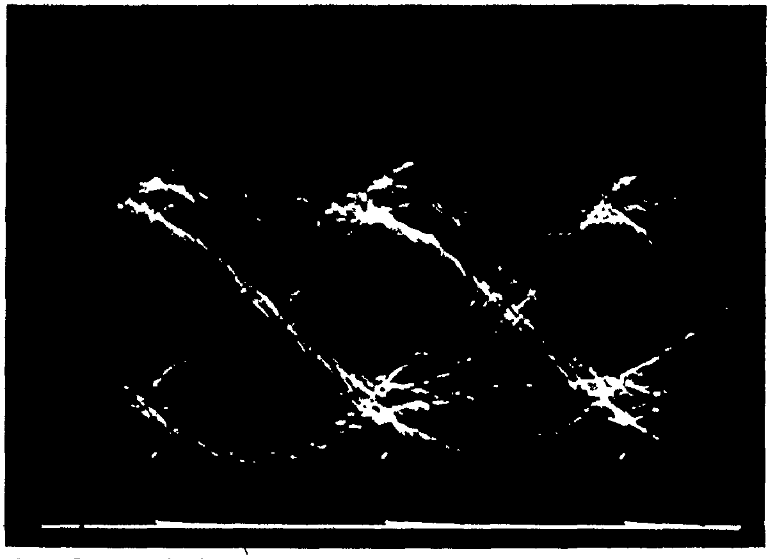
H: 0.5ms/div , V: >0.1V/div & 0.2V/div

Fig.4.27 Random Noise (Upper Trace) and Pseudo-Random Noise (Lower Trace) Generated by the Conventional and Prototype Units, resp.



H: 10µs/div , V: 1V/div & >1V/div

Fig.4.28 Eye Diagram at Receive Filter Input When Random Noise is Added to Transmitted Signal (No Predistortion). 32 kHz Clock Pulses are Shown in the Lower Trace.

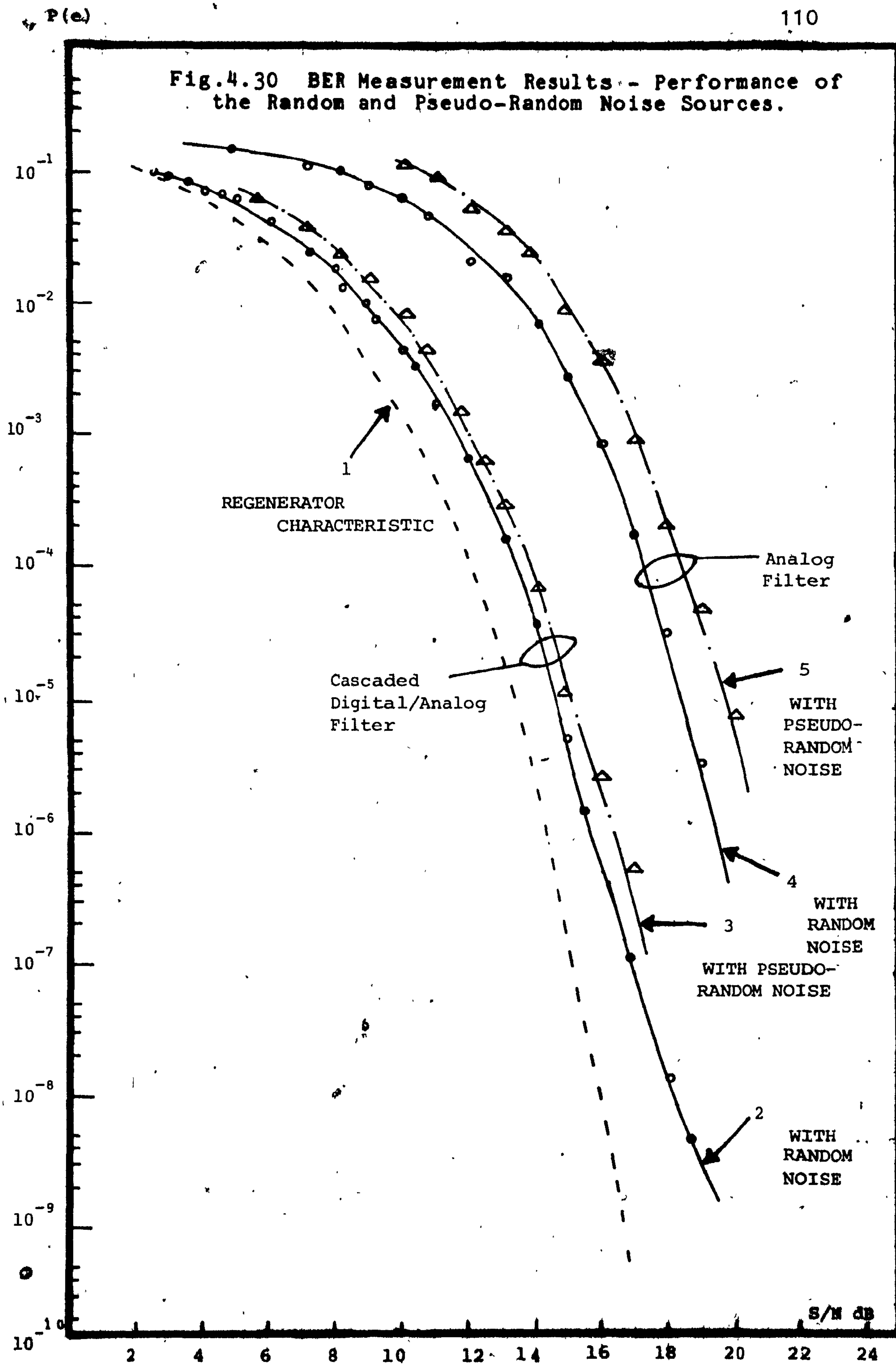


H: 10μs/div , V: 0.1V/div & >1V/div

Fig.4.29 Eye Diagram at Receive Filter Input When Pseudo-Random Noise is Added to Transmitted Signal (No Predistortion). 32 kHz Clock Pulses are Shown in the Lower Trace.

noise with regard to their bandwidths. GR1383 generates white gaussian noise in the band of 20 Hz - 20 MHz, whereas the noise bandwidth of the prototype unit is less than 25 kHz.

Figure 4.30 shows the BER performance of the transmission system when the two types of noise are used separately. The dashed curve 1, again, is the regenerator characteristic when white gaussian noise was used in the test. Curves 2 and 4 are the same curves as in Fig.4.26 but marked 3 and 4, i.e., the performance of the transmit AF with and without the TF cascaded in front of it when the


P(e)
Fig.4.30 BER Measurement Results - Performance of the Random and Pseudo-Random Noise Sources.
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
1
REGENERATOR CHARACTERISTIC
Analog Filter
Cascaded Digital/Analog Filter
5
WITH PSEUDO-RANDOM NOISE
4
WITH RANDOM NOISE
3
WITH PSEUDO-RANDOM NOISE
2
WITH RANDOM NOISE
S/N dB
2
4
6
8
10
12
14
16
18
20
22
24

noise added was white gaussian. Similarly, curves 3 and 5 in Fig.4.30 were obtained in the same order, but the noise being produced by the prototype unit. Hence, these curves provide the evidence that the performance of both noise generators is almost of the same quality, within a 1.0 - 2.0 dB degradation in S/N with the prototype generator.

Chapter 5

CONCLUSION AND SUGGESTED FUTURE RESEARCH

Conventional analog filters when used in digital transmission systems introduce a time-domain distortion in the signal, which is known as the intersymbol-interference. This distortion is caused mainly by the non-linear phase (group-delay) characteristics of the filter, although its amplitude characteristic may be just right for shaping the signal spectrum. An alternative solution to a complex equalization procedure for the analog filter was shown viable. Here, a binary transversal filter is cascaded with the conventional analog filter (at transmit end) so that the signal is predistorted prior to spectral shaping. In this way the size, weight and cost of the composite filter are minimized. The transversal filter is made up of a shift-register, a network of resistors and a summing device (operational amplifier). This digital filter accepts the input non-return-to-zero digital signal, predistorts it according to an established algorithm and passes it on to the analog filter for shaping its spectrum. Experimental results showed that an improvement as high as 4.0 dB could be obtained in the 'eye opening' of the signal at the analog filter output when the digital filter is cascaded to it. The cascaded digital/analog filter was evaluated for its bit error rate performance in a baseband transmission system in an additive white gaussian noise environment. The improvement in signal-to-noise ratio due to the cascade was found to be 4.5 dB at a bit error rate of 1×10^{-4} and 4.0 dB at a bit error rate of 1×10^{-6}.

In evaluating the performance of such filters in a transmission system a random noise source, exhibiting a gaussian amplitude characteristic among its output samples, is often used. Based on a computer simulation results, a simple and low cost noise source was implemented and the behaviour with regard to its output amplitude characteristics was investigated. The noise source consists of a feedback shift-register and a simple resistor-capacitor low pass filter. The feedback circuit was chosen such that the register generated a maximum-length pseudo-random binary sequence, whose properties were discussed at length in the early part of the thesis. Both computer simulation and hardware test results showed that the cut-off frequency of the low pass filter must be set to one-twentieth of the bit rate of the sequence to obtain the desired gaussian noise samples. The aforementioned performance evaluation tests of the cascaded digital/analog filter was repeated using this laboratory-built gaussian noise source. The test results showed a negligible difference in performance between the two noise sources.

<u>Suggested Future Research</u>. The computation of tap-gain coefficients was based on the eqn.(4.3) where the centre sample of the pulse response (of analog filter) was normalized to unity, while forcing all other samples to zero amplitude, simultaneously. Mention was not given, at all, to what happens to pulse amplitude at instants in-between sampling points. Therefore, in computing the optimum tap-gain coefficients the pulse may assume any amplitude at these in-between points. In Fig.4.13 this is obvious because the pulse has assumed an asymmetric shape about the centre. It would be an interesting and a valuable endeavour to obtain the coefficients which would force the pulse to follow a Nyquist shape of a sin(x)/x waveform, but up to a finite duration. This type of a symmetrical pulse would aid in clock recovery at the receiver - an added bonus to the ISI removal. In order to achieve this the shift-register must be clocked at a faster rate than, but some integer multiple of, the bit rate of the digital signal. The eqn.(4.3) must also be modified to include the desired pulse amplitudes between sampling points. The latter can be obtained from the sin(x)/x function.

REFERENCES

Antoniou, A., Digital Filters: Analysis and Design, New York: McGraw-Hill Book Co., 1979.

Bardou, J.P., "RC network prevents power-up latching in a pseudo-random noise generator," Electronic Design, p.142, February 1981.

Barrodale, I. and F.D.K. Roberts, "Solution of an overdetermined System of Equations in the ℓ_1 Norm[F4]," Communications of the ACM, vol.17, no.6, pp.319-20, June 1974.

Bennett, W.R. and J.R. Davey, Data Transmission, New York: McGraw-Hill Book Co., 1965.

Berlekamp, E.R., Algebraic Coding Theory, New York: McGraw-Hill Book Co., 1968.

Bogner, R.E. and A.G. Constantinides, Introduction to Digital Filtering, London: John Wiley and Sons, 1975.

Brind'Amour, A. and K. Feher, "Design and Evaluation of a Convolutional Codec in Additive White Gaussian Noise, Sinusoidal Interference and Intersymbol-Interference Environment," IEEE Transactions on Communications, vol.COM-28, no.3, pp.391-5, March 1980.

Bruton, L.T., RC-Active Circuits: Theory and Design, Englewood Cliffs: Prentice-Hall, Inc., 1980.

Carrefour, A.V., F. Boeri, F. Dubas and D. Esteban, "Synthesis of a transversal digital filter defined in the comxlex plane," Ann. Telecomm., Paris, 28(7-8), pp.297-303, July-August 1973.

Chan, H.C., D.P. Taylor and S.S. Haykin, "Comparative Evaluation of a Digital Modulation Techniques: Simulation Study," CRL Internal Report Series No.CRL-18 Part III, Hamilton, Ontario: Communication Research Laboratory, McMaster University, April 1974.

Cumming, I.G., "Autocorrelation function and spectrum of a filtered pseudo-random binary sequence," Proc. of IEEE, vol.114, pp.1360-2, September 1976.

De Cristofaro, R., "Transversal Filter Design and Application in Satellite Communications," M. Eng. Thesis, Montreal: Concordia University, April 1976.

Douce, J.L. and T.J. Healy, "Evaluation of the amplitude distribution of the quasi-gaussian signals obtained from a pseudo-random noise," IEEE Trans. on Computers, vol.C-18, pp.749-52, August 1969.

Evans, J.V. and T. Hagfois, eds., Radar Astronomy, New York: McGraw-Hill Book Co., 1968.

Feher, K., Digital Communications: Microwave Applications, Englewood Cliffs: Prentice-Hall, Inc., 1981.

Feher, K. and R. de Cristofaro, "Transversal filter design and applications in satellite conmmunications," IEEE Trans. on Commun., vol.COM-24, pp.1262-7, November 1976.

Fiestel, H., W.A. Notz and J.L. Smith, "Some cryptographic techniques for machine-to-machine data communications," Proc. of IEEE, vol.63, pp.1545-54, 1975.

Frequency Devices, Inc., Catalog of Communications Components and Power Supplies, Haverhill, Mass., 1979.

Gellert, W., H. Kustner, M. Hellwich and H. Kastner, eds., The VNR Encyclopedia of Mathematics, New York: Van Nostrand Reinhold Co., 1977.

Golomb, S.W., ed., Digital Communications with Space Applications, Englewood Cliffs: Prentice-Hall, Inc., 1964.

Hampton, R.L.T., "A hybrid analog-digital pseudo-random noise generator," SIMULATION, pp.179-87, March 1965.

Harvey, J.T., "Feedback extends sequence of random-number generator," Electronics Designer's Casebook No.2, New York: McGraw-Hill Book Co., p.46, 1978.

Haykin, S., Comunication Systems, New York: John Wiley and Sons, Inc., 1978.

Henriksson, U., "On a scarmbling property of feedback shift-registers," IEEE Trans. on Commun., vol.COM-20, pp.998-1001, 1972.

Huang, J.C-Y., "On bandwidth efficient spectral shaping methods and digital modulation techniques in linear and non-linear channels," Ph. D. Thesis, Montreal: Concordia University, 1979.

Huelsman, L.P., Active Filters: lumped, distributed, integrated, digital and parametric, New York: McGraw-Hill Book Co., 1970.

Johnson, D.E., J.R. Johnson and H.P. Moore, A Handbook of Active Filters, Englewood Cliffs: Prentice-Hall, Inc., 1980.

Kallman, H.E., "Transversal filters," Proc. of IRE, vol.28, pp.302-10, July 1940.

Kramer, C., "A low frequency pseudo-random noise generator," Electronics Engineering, vol.37, pp.465-7, July 1965.

Lipson, E.D., K.W. Foster and M.P. Walsh, "A versatile pseudo-random noise generator," IEEE Trans. on Instrumentation and Measurements, vol.25, no.2, pp.112-6, June 1976.

Lucky, R.W., "Automatic equalization for digital communication," Bell System Technical Journal, vol.44, no.4, pp.547-88, April 1965.

-------- and H.R. Rudin, "An automatic equalizer for general-purpose communication channels," BSTJ, vol.46, no.9, pp.2179-208, November 1967.

--------, J. Salz and E.J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill Book Co., 1968.

McFadden, J.A., "The probability density of the output of a filter when the input is random telegraph wave," 1959 IRE Convention Record, Part 4, pp.164-9, 1959.

McWilliams, F.J. and N.J.A. Sloane, "Pseudo-random sequences and arrays," Proc. of IEEE, vol.64, no.12, pp.1715-29, December 1976.

--------, The Theory of Error Correcting Codes, Amsterdam: North-Holland Publishing Co., 1978.

Mueller, K.H., "A new approach to optimum pulse shaping in sampled systems using time-domain filtering," BSTJ, vol.52, no.5, pp.723-9, May-June 1973.

Nakamura, K. and Y. Iwadare, "Data scramblers for multilevel pulse sequences," NEC Research and Development, Tokyo, no.26, pp.53-63, July 1972.

Natarajan, S., "Some design and optimization techniques for extending the operating frequency range of active RC filters," Ph. D. Thesis, Montreal: Concordia University, 1979.

Pattengill, G.H., Radar Handbook, M.I. Skolnik, ed., New York: McGraw-Hill Book Co., 1970.

Perera, A.E., K. Feher and M.N.S. Swamy, "Design of cascaded digital and analog filters to meet the requirements on ISI and Bandlimitation in low and medium capacity digital transmission systems," Conference Digest IEEE IEECE, Toronto, pp.126-7, 1979.

Roberts, P.D. and R.H. Davis, "Statistical properties of smoothed maximal length linear binary sequences," Proc. of IEE, vol.113, pp.190-6, January 1966.

Rowe, I.H. and I.M. Kerr, "A broad-spectrum pseudo-random gaussian noise generator," IEEE Trans. on Automatic Control, vol.AC-15, no.5, pp.529-34, October 1970.

Sallen R.P. and E.L. Key, "A practical method of designing RC active amplifiers," IRE Trans. on Circuit Theory, vol.CT-2, pp.74-85, March 1955.

Savage, J.E., "Some simple self-synchronizing digital data scramblers," BSTJ, vol.46, pp.449-87, 1967.

Spilker, J.J., Digital Communication by Satellite, Englewood Cliffs: Prentice-Hall, Inc., 1977.

Stahnke, W., "Primitive binary polynomials," Math Comput., vol.27, pp.977-80, 1973.

Szentirmai, G., ed., Computer Aided Filter Design, Selected Reprints, New York: IEEE Press, 1973.

Temes, G.C. and S.K. Mitra, eds., Modern Filter Theory and Design, New Jersey: John Wiley and Sons, Inc., 1973.

Voelcker, H.B., "Generation of digital signalling waveforms," IEEE Trans. on Commun. Technology, vol.COM-16, no.1, pp.81-93, February 1968.

Zierler, N. and J. Brillhart, "On primitive trinomials (mod 2)," Inform. Cont., vol.13, pp.541-54, 1968 and vol.14, pp.556-9, 1969.

APPENDIX A

MAXIMAL-LENGTH SEQUENCE GENERATION

A.1 Introduction

The number of different maximum-length series that can be obtained from an m-stage shift-register is given by [Hampton, 1965],

$$\frac{\phi(2^m-1)}{m}$$

where $\phi(x)$ is "Euler's Phi Function" that is defined as the number of integers s, such that $0<s<x$ and s is prime to x. For example, for a three-stage generator $m = 3$, $x = 2^3-1 = 7$, and all integers s, $0<s<7$, are 1, 2, 3, 4, 5 and 6; all these are prime to 7, because none of them divides 7 perfectly. Thus, the number of integers s, which satisfy the definition is 6. Therefore,

$$\frac{\phi(2^3-1)}{3} = \frac{6}{3} = 2$$

Hence, a three-stage shift-register can produce two different maximum length sequences, each corresponding to a unique feedback arrangement. The generators No.1 and No.2 in Fig.2.3 show these two arrangements. Generator No.3 is an example of a feedback arrangement that produces a non-maximum length sequence. Such sequences do not in general satisfy the conditions required for pseudo-random sequences as explained in the beginning of Section 2.2 and, thus, are not useful for statistical studies.

As illustrated above, Euler's Phi Function, also known as "Totient Function," can be used to determine how many different maximal-length sequences can be generated by any given number of stages in a feedback shift-register. Section A.2 describes a method of obtaining $\phi(x)$ for any x.

A.2 Euler's Phi Function

The Euler's Phi Function, also known as Euler's Totient Function, $\phi(n)$ is defined as the number of integers not exceeding and relatively prime to n [Gellert et al., 1977].

$$\phi(n) = n \prod_{p|n} (1 - 1/p)$$

over distinct primes p dividing n.

Also $\phi(p) = p-1$, if p is prime number,

$$\phi(p^k) = p^k - p^{k-1} = p^{k-1}(p-1)$$

$\phi(a,b) = \phi(a).\phi(b)$, provided that the greatest common divisor of $(a,b) = 1$.

The rules stated above make it possible to calculate $\phi(m)$ for any m.

<u>Example</u>:

$$\phi(3240) = \phi(2^3).\phi(3^4).\phi(5) = (2^2.1).(3^3.2).4 = 864$$
$$\phi(4095) = \phi(3^2).\phi(5).\phi(7).\phi(13) = (3.2).4.6.12 = 1728$$
$$\phi(262143) = \phi(3^3).\phi(7).\phi(19).\phi(73) = (3^3.2).6.18.72 = 139968$$

Table A.1 Relationship Between Shift-Register Length and Number of Different m-Sequences Generated

m	$2^m-1=x$	Euler's Phi Function, $\phi(x)$	$\phi(x)/m$
2	3	$\phi(3) =$ [illegible]	1
3	7	$\phi(7) = 6$	2
4	15	$\phi(15) = \phi(3).\phi(5) = 8$	2
5	31	$\phi(31) = 30$	6
6	63	$\phi(63) = \phi(3^2).\phi(7) = 36$	6
7	127	$\phi(127) = 126$	18
8	255	$\phi(255) = \phi(3).\phi(5).\phi(17) = 128$	16
9	511	$\phi(511) = \phi(7).\phi(73) = 432$	48
10	1023	$\phi(1023) = \phi(3).\phi(11).\phi(31) = 600$	60
11	2047	$\phi(2047) = 2046$	186
12	4095	$\phi(4095) = \phi(3^2).\phi(5).\phi(7).\phi(13) = 1728$	144
13	8191	$\phi(8191) = 8190$	630
14	16383	$\phi(16383) = \phi(3).\phi(43).\phi(127) = 10584$	756
15	32767	$\phi(32767) = \phi(7).\phi(31).\phi(151) = 27000$	1800
16	65535	$\phi(65535) = \phi(3).\phi(5).\phi(17).\phi(257) = 32768$	2048
17	131071	$\phi(131071) = 131070$	7710
18	262143	$\phi(262143) = \phi(3^3).\phi(7).\phi(19).\phi(73) = 139968$	7776
19	524287	$\phi(524287) = 524286$	27594
20	1048575	$\phi(1048575) = \phi(3).\phi(5^2).\phi(11).\phi(31).\phi(41) = 480000$	24000

<u>Note</u>: 1. m = Shift-Register Length

2. $2^m-1 = x$ = number of different sequences, excluding all-zero, generated by the shift-register.

3. $\phi(2^m-1)/m$ = number of different maximum length series obtainable from the m-stage shift-register.

Table A.2 Primitive Polynomials for Degree $m \leq 40$

m	h(x)	m	h(x)
1	$x+1$	21	$x^{21} + x^2 + 1$
2	$x^2 + x + 1$	22	$x^{22} + x + 1$
3	$x^3 + x + 1$	23	$x^{23} + x^5 + 1$
4	$x^4 + x + 1$	24	$x^{24} + x^4 + x^3 + x + 1$
5	$x^5 + x^2 + 1$	25	$x^{25} + x^3 + x + 1$
6	$x^6 + x + 1$	26	$x^{26} + x^8 + x^7 + x + 1$
7	$x^7 + x + 1$	27	$x^{27} + x^8 + x^7 + x + 1$
8	$x^8 + x^6 + x^5 + x + 1$	28	$x^{28} + x^3 + 1$
9	$x^9 + x^4 + 1$	29	$x^{29} + x^2 + 1$
10	$x^{10} + x^3 + 1$	30	$x^{30} + x^{16} + x^{15} + x + 1$
11	$x^{11} + x^2 + 1$	31	$x^{31} + x^3 + 1$
12	$x^{12} + x^7 + x^4 + x^3 + 1$	32	$x^{31} + x^{28} + x^{27} + x + 1$
13	$x^{13} + x^4 + x^3 + x + 1$	33	$x^{33} + x^{13} + 1$
14	$x^{14} + x^{12} + x^{11} + x + 1$	34	$x^{34} + x^{15} + x^{14} + x + 1$
15	$x^{15} + x + 1$	35	$x^{35} + x^2 + 1$
16	$x^{16} + x^5 + x^3 + x^2 + 1$	36	$x^{36} + x^{11} + 1$
17	$x^{17} + x^3 + 1$	37	$x^{37} + x^{12} + x^{10} + x^2 + 1$
18	$x^{18} + x^7 + 1$	38	$x^{38} + x^6 + x^5 + x + 1$
19	$x^{19} + x^6 + x^5 + x + 1$	39	$x^{39} + x^4 + 1$
20	$x^{20} + x^3 + 1$	40	$x^{40} + x^{21} + x^{19} + x^2 + 1$

A.2 On Obtaining Maximal-Length

A mathematical technique for obtaining a maximum-length sequence consists of viewing each state of the m-stage shift-register as an m-dimensional vector and the shift-register/modulo-2 adder system as a linear operator, producing the successive states of the m-dimensional vector. Such an operation may be represented by a [m x m] matrix **X**. The first row of **X** corresponds to the first stage of the register, the second row to the second stage, etc. The same is true for the m-columns, i.e., the first column represents the first stage, etc. Each element of X is either a logic 0 or a logic 1. A logic 1 in any position indicates that the flip-flop stage denoted by that column drives the stage denoted by the particular row. Otherwise, a logic 0 element exists. Considering generator No.1 in Fig.2.3 of Section 2.2.1, X may be constructed by observing that stage 1 is fed by stages 2 and 3; therefore, in the first row, a logic 1 is entered in columns 2 and 3. Stage 2 is fed by stage 1 only, and so, in row 2 a logic 1 is placed in column 1 only. Again, stage 3 is driven by stage 2; hence, in row 3 a logic 1 is placed in column 2. This completes the **X** matrix for generator No.1 as follows:

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Similarly, for generators No.2 and No.3 the defining matrices are,

$$X = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

respectively. The diagonal below the main diagonal will always consist of a series of 1s, while the first row of the matrix represents feedback coefficients. Generalizing, the **X** matrix for m stages may be written:

$$X = \begin{bmatrix} C_1 & C_2 & \ldots & C_i & \ldots & C_m \\ 1 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \ldots & 0 \\ . & . & 1\ldots & & \ldots & \\ . & . & \ldots 1. & & \ldots & \\ . & . & \ldots & & 1\ldots & \\ 0 & 0 & \ldots & 0 & \ldots 1 & 0 \end{bmatrix}$$

where C_i represent the feedback coefficients (0 or 1), $i = 1, \ldots, m$.

The characteristic polynomial of **X** is thus,

$$\det\{X-\lambda I\} \equiv \begin{vmatrix} C_1-\lambda & C_2 & C_3 & \ldots & C_i & \ldots C_{m-1} & C_m \\ 1 & -\lambda & 0 & \ldots & 0 & \ldots & 0 & 0 \\ 0 & 1 & -\lambda & \ldots & 0 & \ldots & 0 & 0 \\ . & . & 1 & \ldots & . & \ldots & . & . \\ . & . & . & \ldots & . & \ldots & -\lambda & . \\ 0 & 0 & 0 & \ldots & 0 & \ldots & 1 & -\lambda \end{vmatrix}$$

$$\begin{aligned}\det\{\mathbf{X}-\lambda\mathbf{I}\} &= (-\lambda)^{m-1}(C_1-\lambda) - (-\lambda)^{m-2}C_2 + (-\lambda)^{m-3}C_3 \\ &\quad -+ \ldots + (-\lambda)^{m-m}C_m \\ &= (-\lambda)^m(1 - C_1/\lambda - C_2/\lambda^2 - C_3/\lambda^3 - \ldots \\ &\quad - C_m/\lambda^m) \\ &= \left(\frac{-1}{\theta^m}\right)^m [1 - \Sigma_{i=1}^m C_i\theta^i]\end{aligned}$$

where $\theta = 1/\lambda$.

The term

$$\left(\frac{-1}{\theta^m}\right)^m [1 - \Sigma_{i=1}^m C_i\theta^i]$$

when equated to zero, is defined as "the characteristic equation of the shift-register generator." A necessary, but not sufficient, condition for the period of shift-register to be of maximum length is that its characteristic equation be irreducible. In Fig.2.3 of Section 2.2.1 the characteristic equations of both generators No.1 and No.2 are irreducible, but not that of generator No.3.

A.3 The Primitive Polynomials

In the previous section it was established that a maximal length sequence from a feedback shift-register generator could be obtained only if the characteristic equation $[1-\Sigma_{i=1}^{m} C_i\theta^i] = 0$ is irreducible. The determinant $\det\{\mathbf{X}-\lambda\mathbf{I}\}$ can be developed to obtain a polynomial of degree m in λ. This polynomial is called the "characteristic polynomial" corresponding to $\mathbf{X}$.

The characteristic polynomial of degree m in λ corresponding to $\mathbf{X}$ when irreducible is termed a "primitive polynomial". Conversely, to construct a PRBS of length $n = 2^m-1$ (maximal length) one needs a primitive polynomial h(x) of degree m. For example,

$$h(x) = x^4 + x + 1 \tag{A.1}$$

is a primitive polynomial of degree m = 4. This polynomial, specifies the feedback configuration for a 4-stage shift-register as shown in Fig.A.1. Here, if the register contains at time i,

$$a_{i+3},\ a_{i+2},\ a_{i+1} \text{ and } a_i$$

then, at time (i+1) it contains,

$$a_{i+4} = a_{i+1} \oplus a_i,\ a_{i+3},\ a_{i+2} \text{ and } a_{i+1}$$

where $\oplus$ denotes modulo-2 addition. Thus the feedback shift-register generates an infinite sequence

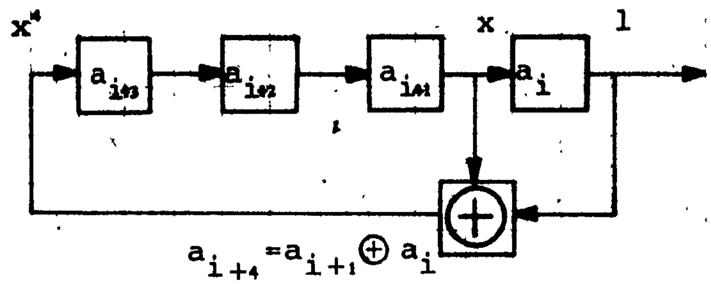


Fig.A.1 Feedback Shift-Register corresponding to x^4+x+1.

$a_0a_1a_2 \ldots a_i \ldots$ which satisfies the recurrence relationship,

$$a_{i+4} = a_{i+1} \oplus a_i, \quad i = 0, 1, \ldots \tag{A.2}$$

The shift-register must be started-up and, therefore, the initial values $a_0, a_1, \ldots, a_{m-1}$ must all be specified.

Definition: A primitive polynomial h(x) is a polynomial of degree m for which $a_0a_1a_2\ldots$ has period (2^m-1) for some starting state.

It has been proved [Berlekamp, 1968; MacWilliams and Sloane, 1978] that there exists primitive polynomials of degree m for every m. Table A.2 is an extraction (for $m \leq 40$; sufficient to generate sequences of period up to $2^{40}-1 \cancel{\cong 10^{12}}$) of the table of primitive polynomials ($m \leq 168$) given by Stahnke [1973]. Primitive polynomials of much higher degree have been found and reported by Zierler and Brillhart [1968,1969].

It follows that if $h(x)$ is a primitive polynomial of degree m, the shift-register goes through all (2^m-1) distinct non-zero states before repeating, and produces an output sequence $a_0a_1a_2\ldots$ of period (2^m-1). Any segment $a_ia_{i+1}\ldots a_{i+2^m-2}$ of length (2^m-1) is called a "pseudo-random sequence". There are (2^m-1) different pseudo-random sequences (setting $i = 0, 1, \ldots, 2^m-2$ in the above expression).

<u>NOTE</u>: It is possible, however, to attain period 2^m (instead of 2^m-1) using a nonlinear shift-register. The corresponding output sequence is called "de Bruijn cycle" [MacWilliams and Sloane, 1976].

APPENDIX B

REALIZATION OF AN EIGTH-ORDER BUTTERWORTH ACTIVE ANALOG FILTER

The eighth-order Butterworth analog LPF was realized by cascading four second-order LPF sections that were independently synthesized to yield the desired charactersistics in the overall filter. The second-order transfer function used is of the form [Temes and Mitra, 1955],

$$H(s) = \frac{H\omega_0^{\ 2}}{s^2+(\omega_0/Q)s+\omega_0^{\ 2}} \tag{B.1}$$

where Q is known as the pole pair of the network, ω_0 the cut-off frequency of the passband, and H the gain factor of the network. Among the wide variety of second-order filter configura- tions is a simple one due to Sallen and Key [1955], which is well known by filter designers. It consists of an amplifier, two capacitors and two resistors as shown in Fig.B-1. The voltage transfer function of this configuration can easily be deduced as,

$$\frac{V_o}{V_i} = \frac{\dfrac{KG_1G_2}{C_1C_2}}{s^2+\left\{\dfrac{G_1+G_2}{C_1}+\dfrac{(1-K)G_2}{C_2}\right\}s+\dfrac{G_1G_2}{C_1C_2}} \tag{B.2}$$

Comparing eqns.(B.1) and (B.2), it is seen that

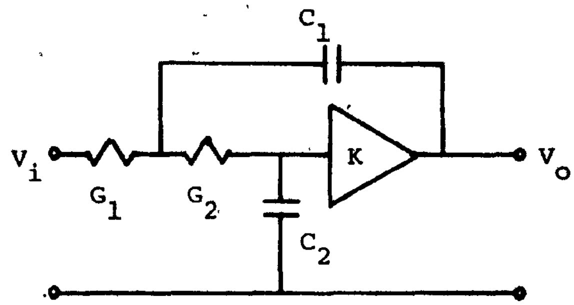


Fig.B-1 Sallen and Key configuration of a 2nd-order LPF.

$$\omega_0^2 = \frac{G_1G_2}{C_1C_2} \tag{B.3}$$

$$Q = \frac{\sqrt{C_1C_2G_1G_2}}{C_2(G_1+G_2)+(1-K)C_1G_2} \tag{B.4}$$

The synthesis can be simplified greatly if ω_0 is normalized to 1 rad/sec. To move ω_0 later, perhaps, to M rad/sec, all capacitances are divided by M so that all capacitive reactances are kept invariant through the frequency translation. Thus,

$$\frac{G_1G_2}{C_1C_2} = 1 \tag{B.5}$$

and,

$$\frac{G_1+G_2}{C_1} + \frac{(1-K)G_2}{C_2} = \frac{1}{Q} \tag{B.6}$$

One can further simplify the synthesis by using a unity gain amplifier, i.e., setting K = 1, and normalizing the resistors to have a value of 1 ohm each. Equations (B.5) and (B.6), therefore, reduce to,

$$C_1 = 2Q \tag{B.7}$$

$$C_2 = 1/2Q \tag{B.8}$$

A very important criterion that must be taken into consideration when designing filters is the sensitivity figure. The sensitivity is an estimate of the effect of parameter variations on the network characteristics. The two most pertinent sensitivity figures applicable to the work here are the pole pair Q-sensitivity and the pole frequency sensitivity.

The sensitivity of the pole pair Q with respect to a network parameter x is defined as,

$$S_x^Q = \frac{d(\ln Q)}{d(\ln x)} = \frac{x}{Q} \cdot \frac{dQ}{dx} \tag{B.9}$$

Thus S_x^Q gives the incremental change in Q due to an incremental change in x.

The sensitivity of the pole frequency ω_n is defined similarly as,

$$S_x^{\omega_n} = \frac{x}{\omega_n} \cdot \frac{d\omega_n}{dx} \tag{B.10}$$

Substituting with the parameters in eqn.(B.2) for x in eqns.(B.9) and (B.10), one obtains

$$S_{G1}^{Q} = S_{G2}^{Q} = 0 \tag{B.11}$$

$$S_{C1}^{Q} = - S_{C2}^{Q} = 1/2 \tag{B.12}$$

$$S_{\mu}^{Q} \simeq 2Q^2/\mu \tag{B.13}$$

where μ is the open-loop gain of the operational amplifier that is used to realize the unity gain amplifier mentioned before. The passive parameter sensitivities are very low. The active parameter sensitivity is also low provided $Q^2 << \mu$. Hence, this restricts the usefulness of this structure to low Q filter realizations.

The resistance values were previously set to 1 ohm to simplify the synthesis. They can be scaled later without upsetting the cut-off frequency ω_0 which is preset to 1 rad/sec. This is done by multiplying all resistances by p, an arbitrary scale factor, and dividing all capacitances by p, simultaneously.

The n^{th}-order Butterworth low pass filter response function, H(s), is given by,

$$H(s) = \frac{1}{1+a_1s+a_2s^2+\ \dots\ +a_ns^n} \tag{B.14}$$

where the denominator polynomials are referred to as "Butterworth Polynomials" often found tabulated in text books [Van Valkenburg, 1960]. Table B.1 shows these polynomials for up to n = 8.

Table B.1 Butterworth Polynomials

n	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	1.0000							
2	1.4142	1.0000						
3	2.0000	2.0000	1.0000					
4	2.6131	3.4142	2.6131	1.0000				
5	3.2361	5.2361	5.2361	3.2361	1.0000			
6	3.8637	7.4641	9.1416	7.4641	3.8637	1.0000		
7	4.4940	10.0978	14.5918	14.5918	10.0978	4.4940	1.0000	
8	5.1258	13.1371	21.8462	25.6884	21.8462	13.1371	5.1258	1.0000

In order to obtain a satisfactory steep roll-off of stop-band frequencies, an 8^{th}-order Butterworth filter was chosen as the bandlimiting filter. Using Table B.1, the transfer function of an 8^{th}-order Butterworth LPF is,

$$H(s) = \frac{1}{(1+5.1258s+13.1371s^2+21.8462s^3+24.6884s^4+ 21.8462s5+13.1371s^6+5.1258s^7+s^8)} \quad \text{(B.15)}$$

Given an n^{th}-order Butterworth polynomial, the first step is to compute the complex roots of the polynomial. These roots may be found tabulated in texts such as shown in Table B.2, or may be computed with the aid of a computer program.

Using Table B.2, the denominator of right hand side of eqn.(B.15) could be factorized as,

$$H(s) = \frac{1}{\begin{array}{c}\{(s+0.9009689+j0.4338837)(s+0.9009689-j0.4338837) \\ (s+0.5555702+j0.8314696)(s+0.5555702-j0.8314696) \\ (s+0.8314696+j0.5555702)(s+0.8314696-j0.5555702) \\ (s+0.9807853+j0.1950903)(s+0.9807853-j0.1950903)\}\end{array}}$$

Re-structuring the above equation,

Table B.2 Poles of Butterworth Filters

n	Complex Poles
1	-1.0000000
2	-0.7071068 ± j0.7071068
3	-1.0000000
	-0.5000000 ± j0.8660254
4	-0.3826834 ± j0.9238795
	-0.9238795 ± j0.3826834
5	-1.0000000
	-0.3090170 ± j0.9510565
	-0.8090170 ± j0.5877852
6	-0.2588190 ± j0.9659258
	-0.7071068 ± j0.7071068
	-0.9659258 ± j0.2588190
7	-1.0000000
	-0.2225209 ± j0.9749279
	-0.6234989 ± j0.7818315
	-0.9009689 ± j0.4338837
8	-0.1950903 ± j0.9807853
	-0.5555702 ± j0.8314696
	-0.8314696 ± j0.5555702
	-0.9807853 ± j0.1950903
9	-1.0000000
	-0.1736482 ± j0.9848078
	-0.5000000 ± j0.8660254
	-0.7660444 ± j0.6427876
	-0.9396926 ± j0.3420201
10	-0.1564345 ± j0.9876883
	-0.4539905 ± j0.8910065
	-0.7071068 ± j0.7071068
	-0.8910065 ± j0.4539905
	-0.9876883 ± j0.1564345

$$H(s) = \frac{1}{\{(s^2+1.1114s+1)(s^2+1.9614s+1)(s^2+0.3901s+1)(s^2+1.6628s+1)\}} \qquad \text{(B.16)}$$

Therefore, the 8^{th}-order function is a cascade of four second-order functions. Each second-order function can now be implemented independently. It must also be noted that in eqns.(B.15) and (B.16) the cut-off frequency of the 8^{th}-order filter, as well as that of individual second-order sections, is normalized to 1 rad/sec.

Rewriting eqn.(B.16) such that,

$$H(s) = H_1(s).H_2(s).H_3(s).H_4(s) \tag{B.17}$$

where,

$$H_1(s) = \frac{1}{s^2+1.1114s+1} \tag{B.18}$$

$$H_2(s) = \frac{1}{s^2+1.9614s+1} \tag{B.19}$$

$$H_3(s) = \frac{1}{s^2+0.3901s+1} \tag{B.20}$$

$$H_4(s) = \frac{1}{s^2+1.6628s+1} \tag{B.21}$$

Comparing each of the eqns.(B.18) through (B.21) with eqns.(B.1) through (B.8) the values of capacitances and resistances of every section can be computed, for normalized frequency of ω_0, which is 1 rad/sec, as shown in Table B.3.

As mentioned earlier the resistor and capacitor values may now be scaled by any arbitrary value and the filter cut-off frequency ω_0 may be denormalized to any desired

Table B.3 Normalized Values of Capacitance and Resistance of Each Second-Order Section in Eqns.(B.18 to B.21)

SECTION NO.	FREQUENCY [rad/sec] ω_0	GAIN OF AMP.	ω_0/Q	Q	RESISTANCE [ohms] $R_1(=R_2)$	CAPACITANCE [farads] C_1	C_2
1	1	1.0	1.1114	0.8998	1.0	1.7995	0.5527
2	1	1.0	1.9614	0.5098	1.0	1.0197	0.9807
3	1	1.0	0.3901	2.5634	1.0	5.1269	0.1951
4	1	1.0	1.6628	0.6014	1.0	1.2028	0.8314

value $2\pi f_0$, where f_0 is the cut-off frequency in Hertz. The resistor values of sections 1, 2 and 4 were arbitrarily chosen as 4700 ohms, while those of section 3 as 2700 ohms. The cut-off frequency was de-normalized to 16 kHz so that this filter may be used to bandlimit a 32 kb/s (in baseband) random NRZ signal. Thus, for scaling, resistances were multiplied and capacitances were divided by 4700 in sections 1, 2 and 4; and by 2700 in section 3. In order to de-normalize ω_0, all capacitances were divided by 32000π. These de-normalized and scaled values for each section are shown in Table B.4. Cascading the four sections using the computed values given in Table B.4, the 8^{th}-order maximally-flat filter is obtained as shown in Fig.B.2. The exact values of capacitors and resistors computed cannot, however, be realized in practice. Therefore, nominal values that were closest to the computed ones were chosen. These

Table B.4 Denormalized and Scaled values of Capacitance and Resistance for $\omega_0 = 32000\pi$ rad/sec

SECTION NO.	RESISTANCE [ohm] R_1	R_2	CAPACITANCE [nanofarad] C_1	C_2
1	4700	4700	3.8287	1.1761
2	4700	4700	2.1581	2.0756
3	2700	2700	18.8880	0.7186
4	4700	4700	2.5455	1.7596

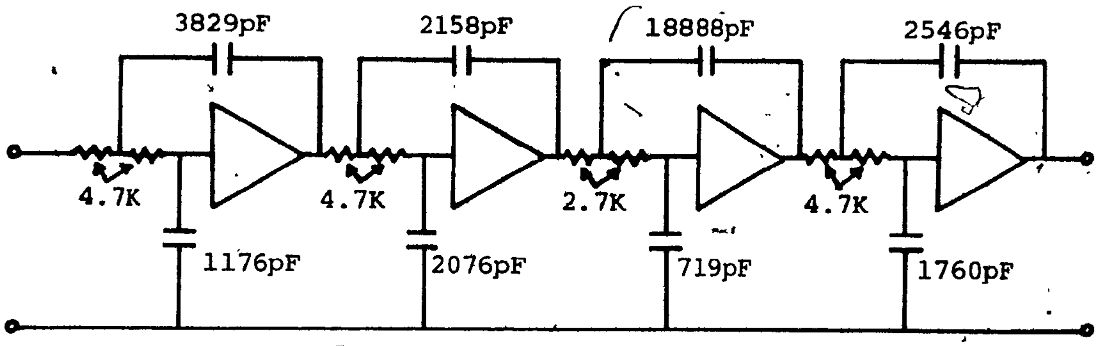


Fig.B.2 Cascaded 4 Sections of 8^{th}-Order LPF with a 3 dB cut-off at 16 kHz.

components were, then, tuned to obtain the desired filter characteristics. This tuning process was done separately on each individual section by measuring the attenuation at 16 kHz. The final configuration of the 8^{th}-order filter tuned for a 3 dB cut-off frequency of 16 kHz is shown in Fig.4.1 in Chapter 4.

APPENDIX C

COMPUTER SIMULATION

C.1. Introduction

In digital computer simulations the signals must be available in the discrete form. Also, the filter transfer function must be available sampled in the frequency domain. There exists a definite relationship between time-domain sampling and frequency-domain sampling. In this simulation, the digital signal is sampled in the time-domain where as the filter transfer function is sampled in the frequency-domain. The signal is converted from one domain to the other through Fast Fourier Transformation.

The programs and associated subroutines for the simulation have been extracted from the Report No.CRL-18 Part III of Communications Research Laboratory of McMaster University, Hamilton, Ontario [Chan et al., 1974]. Most of these subroutines were modified, however, and new ones were written to suit the needs in this work. Some important parameters of the simulation model, which the user must be aware of when using these programs, may be described as follows.

The signal and the transfer function of the filter are sampled and represented in vectors. Each pair of samples, starting from the first element in a vector (array), represents the real and the imaginary parts of a complex sample. Thus, if a signal element is sampled 16 times (LSAMPL = 16) the element is actually represented by 32 samples of which the odd samples are real parts and the even are imaginary parts of the signal element. The long pseudo-random digital signal stream is processed, section by section, to save the memory space in the program. In a 7-stage (JLAST = 7) shift-register that generates the pseudo-random data the number of symbols generated is 2^7-1=127 (NSYMB = 127). Owing to the limitation of computer memory size, these 127 symbols are sectioned into 11 loops (LOOPM = 11) with each loop containing 12 symbols (KKK = 12). When a symbol is sampled 16 times each section is, then, represented by 192 complex samples, which actually occupy 384 positions of the array. Because of the delay inherent in filters the first position in the array does not necessarily contain the first sample of a filtered signal section. This sample may be delayed and positioned somewhere in the array, accountable only by the characteristics of the particular filter. All other samples, in effect, are also delayed by the same amount. Therefore, the array into which the signal is initially loaded must contain at its end an empty space that is large enough to accommodate the spilt-over signal samples. Hence, a 512-point array (LDIM = 512) would contain 384 samples of a

12-symbol signal section and 128 empty points (LLT = 128) for spill over. The filtering action and sectioning/ reconstructing of data are depicted in Figs.C.1 and C.2,

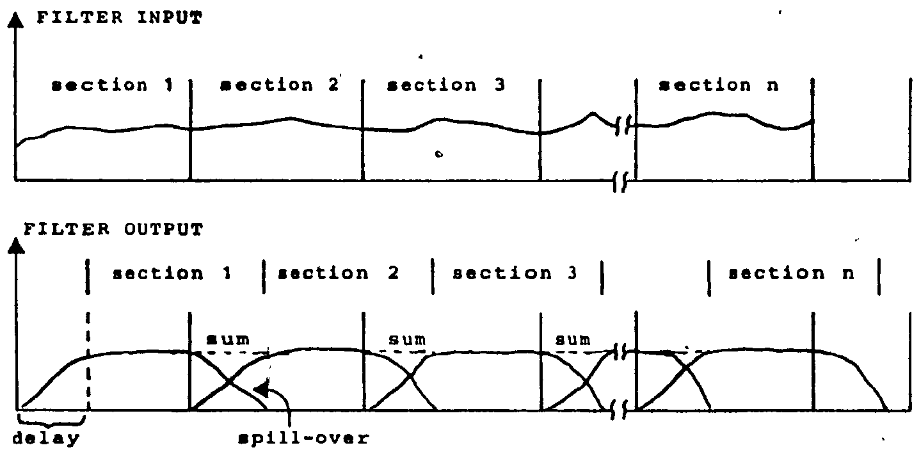


Fig.C.1 Illustration of Filtering Action on Signal Sections.

respectively.

The delay caused by the filter is computed through subroutine BISINC and is given as NSTART. Thus, NSTARTth position of the array DATA (or DATA1) would contain the first sample of a signal section when filtered. At the beginning of the program, a nominal value (LLTNOM) for spill-over is specified. The subroutine CALCON, then, computes other constants and assigns a computed value for LLT, the actual spill-over space. Most filters do not have a delay of more than 50 points of the signal array. LLTNOM may, therefore, be specified as 50. LLT will, in this case, have a computed value of 64. However, the program may have

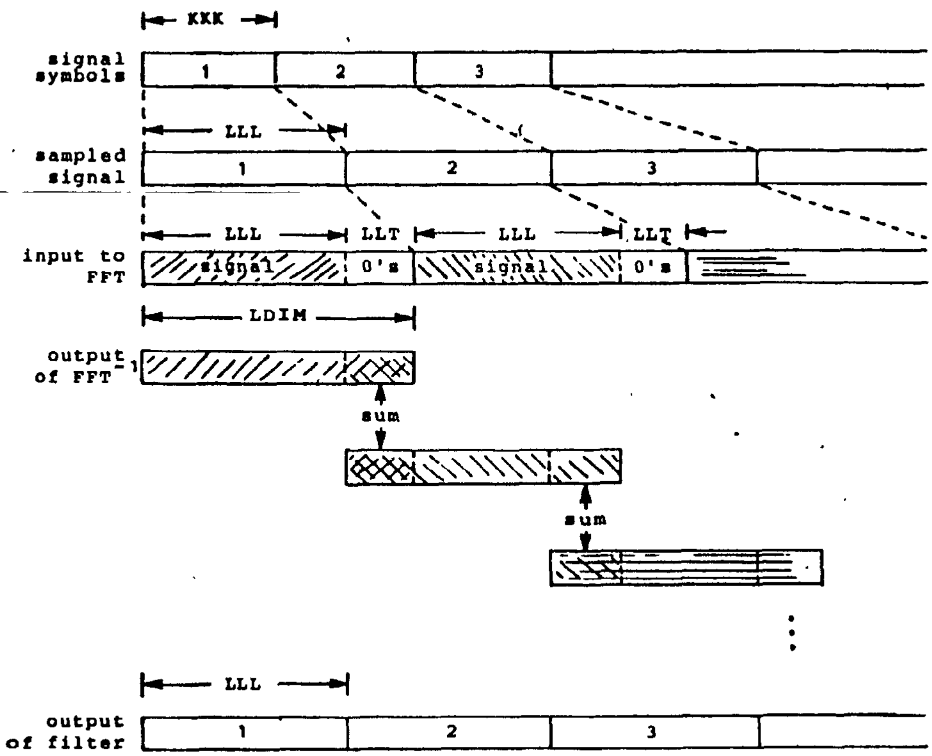


Fig.C.2 Illustration of Sectioning and Reconstructing the Continuous Signal Stream.

to be re-run if NSTART is found to exceed LLT, as printed on output, with a re-specified value for LLTNOM that is larger than NSTART. Letting LLT stay less than NSTART causes the filtered signal to be truncated before the end of each section. During reconstruction of the continuous PRBS stream, some information is, therefore, unavailable resulting discontinuities in the filtered signal. The signal is, then, useless. A side effect of making LLT larger is, of course, that less space is now available in the array for the signal for a given length of LDIM. This increases the number of loops required to process NSYMB symbols of the signal stream; hence, a long processing time.

C.2 Programs and Subroutines

There are two major programs that were used in this work, viz, NOISGEN to simulate gaussian noise by filtering a PRBS stream and SIMULA to simulate the eye diagram at output of a transmit filter in a baseband digital communications system. These two FORTRAN programs correspond to the description given in Chapters 2 and 4. There are three other minor programs that may be used in the event that all the necessary input data for SIMULA is unavailable. If the filter is not assumed theoretical but practical, the polynomial functions of its amplitude and phase characteristics must first be obtained. Program CURVFIT can be used for this purpose (up to a 20th degree) by supplying measured characteristics. In the theoretical case, the polynomials may be computed or found in text books as described in Chapter 4. When equalization of the filter is effected through a transversal filter the optimum tap-gain coefficients can be obtained by using the program PEAKDIS. In order to run it, a sampled pulse response of the filter must be known. In the theoretical case, program PULSE may be used to compute such samples by providing the polynomials of the filter as input data. In the practical case the samples of an isolated pulse response of the filter must be obtained from a CRO display as explained in Chapter 4.

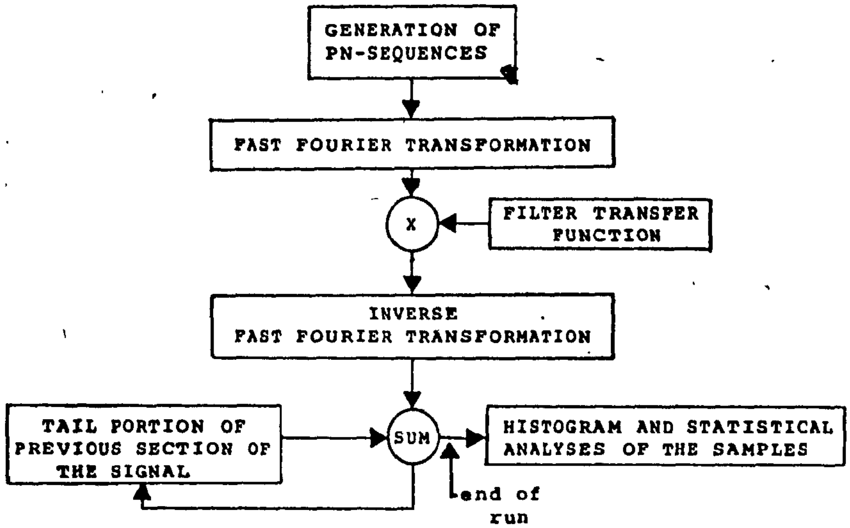


Fig.C.3 Flow Chart for Gaussian Noise Simulation.

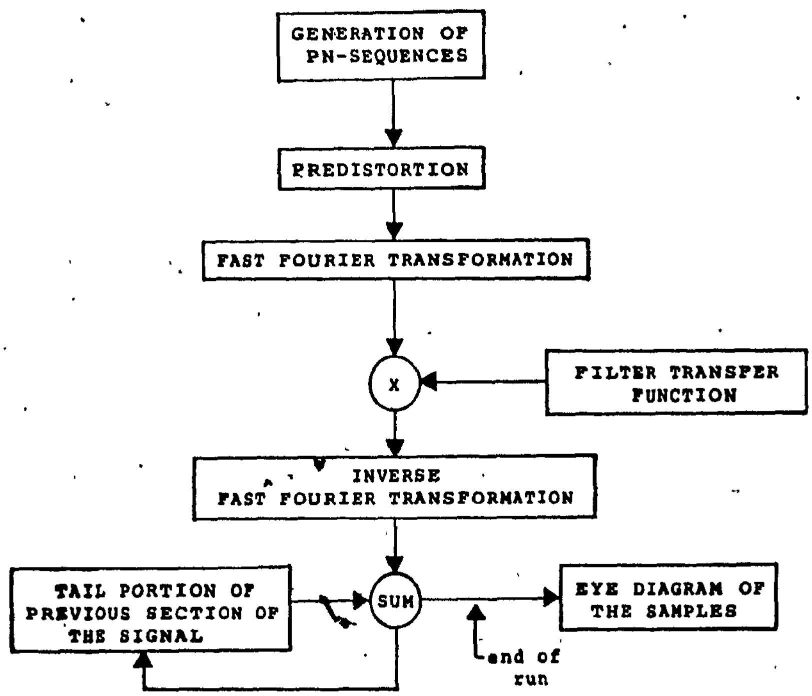


Fig.C.4 Flow Chart for Eye Diagram Simulation.

Figures C.3 and C.4 show Flow Charts that summarize the functions of major programs NOISGEN and SIMULA. Table C1 is a summary of all programs described above. Fig.C.5 shows a Flow Chart of the procedure that must be followed, in order to run SIMULA.

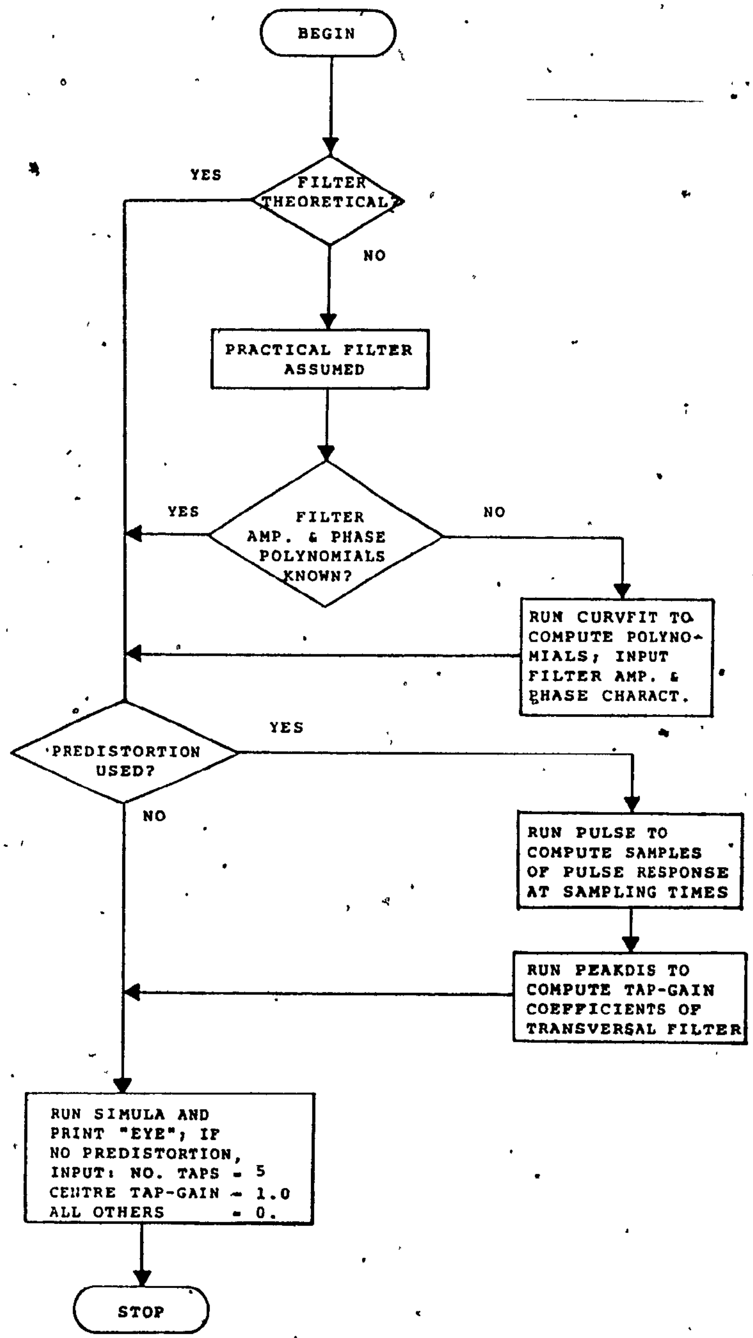


Fig.C.5 Flow Chart for the Simulation Procedure.

Table C.1 Summary of Computer Programs

Item	Name	Description
1	CURVFIT	Computes polynomials of up to a 20^{th} degree equation that fits a given set of data points.
2	NOISGEN	Simulates pseudo-random noise by filtering PRBS using a prescribed filter. The results are printed in a histogram and analyzed for statistical properties.
3	PEAKDIS	Computes tap-gain coefficients of a transversal filter that is used to equalize a given transmit filter in baseband.
4	PULSE	Simulates pulse response of a prescribed filter.
5	SIMULA	Simulates and prints the eye diagram of a filtered pseudo-random signal in baseband with or without predistorting the signal using a transversal filter.

The subroutines used for the simulation are as follows:

ADTAIL1	ADTAIL2	BEMSON	BISINC	CAL	CALCON
DELAY	EYE	FFT2C	FILTER	GDATA	HISTOG
IDFIL	LOAD	L1	MEASFIL	MINV	MULTR
ORDER	PLOT	POLRBW	PREDIST	RCOSMOD	SCUTF6
THEOFIL	TRANSF	USHIUT			

Table C.2 is a summary of the functions of all subroutines named above. It also indicates the programs (numbers refer to items in Table C.1) that utilize individual subroutines.

Table C.2 Subroutines and their Functions

Subrtn.	Programs	Function
ADTAIL1	2,5	Adds the head portion of a section of filtered signal to the tail portion of previous section to reconstruct the continuous stream. These portions are, then, replaced by their sum.
ADTAIL2	5	Same as in ADTAIL1 but for predistorted signal (through transversal filter) prior to filtering.
BEMSON	2	Computes statistical parameters of filtered signal samples. This subroutine is accessed from IMSLIBrary of Con. U. Computer Centre.
BISINC	2,4,5	Determines the starting position of signal samples in the array after passing through the filter.
CAL	3	Initializes the parameters for subroutine L1.
CALCON	2,4,5	Computes the necessary constants for the simulation.
DELAY	2,5	Compensates for the delay in the signal caused by predistortion.
EYE	5	Prints the eye diagram of a signal array.
FFT2C	2,4,5	Performs FFT and inverse FFT on a complex array of signal. This subroutine is accessed from IMSLIB.
FILTER	2,4,5	Filters the signal array using FFT2C. Signal is transformed from time-domain to frequency-domain, multiplied by filter transfer function, and the processed signal is transformed back to timedomain.
GDATA	1	Generates independent variables to a specified m^{th} power ($m \leq 20$) and computes mean, standard deviation and correlation coefficients.
HISTOG	2	Sorts out data samples within a specified amplitude range, counts the frequency of occurrence, and then computes mean, standard deviation of the samples. The computed values are used to obtain the corresponding gaussian (or normal) probability density function.

Table C.2 (continued)

Subrtn.	Programs	Function
IDFIL	4,5	Determines the complex transfer function of an ideal low pass filter.
LOAD	2,5	Generates PRBS and loads them into a specified signal array.
L1	3	Computes an L1 solution to an overdetermined system of linear equations through a modified simplex method.
MEASFIL	4,5	Computes the complex transfer function of a filter whose measured amplitude and phase characteristics are given.
MINV	1	Inverts a matrix using standard Gauss-Jordan method.
MULTR	1	Performs a multiple linear regression analysis for a dependent variable and a set of independent variables.
ORDER	1	Constructs a subset matrix of intercorrelations among independent variables and a vector of intercorrelations of independent variables from a given larger matrix of correlation coefficients.
PLOT	1	Plots several cross-variables versus a base variable.
POLRBW	1	Performs (i) reading the problem parameters for a polynomial regression, (ii) calling subroutines to perform analysis, and (iii) printing and plotting the results.
PREDIST	5	Predistorts a signal array using a transversal filter of specified number of taps and tap-gain coefficients.
RCOSMOD	5	Computes the complex tranfer function of a raised cosine filter modified with x/sin(x) function.
SCUTF6	5	Computes the complex transfer function of a sharp cut-off filter with a 6 dB attenuation at Nyquist frequency.
THEOFIL	2,4,5	Computes the complex transfer function of a theoretical filter when filter order and polynomials are specified.
TRANSF	2,4,5	Transforms a real array of data into a complex array of half size and vice-versa.
USHIUT	2	Prints two histograms (superimposed) on a given amplitude range. This subroutine is accessed from IMSLIB.

C.3.a Decription of SIMULA

Each section of the signal is processed according to the flow chart as shown in Fig.C.4. Two arrays DATA and DATA1 contain signal sections. TAPE1 and TAPE3 are used for temporary storage of intermediate data. The transfer function of the filter is computed and stored in the array TF. TAPE2 is used to store permanently the processed signal, section by section, at the end of each loop of the run. Arrays NX and TAPGAIN contain, respectively, the contents of the shift-register that generates PRBS and the tap-gain coefficients of transversal filter that equalizes the spectral shaping analog filter. Five logical switches SW are used for optional printing of intermediate data during the run.

The operation of the program could be best described with the aid of a flow chart as shown in Fig.C.6. Each section of the signal is loaded into either DATA or DATA1 as necessary by subroutine LOAD. The signal is, then, predistorted by PREDIST. When two consecutive predistorted sections of the signal are available the tail portion of the first section is added to the head portion of the next. The latter is stored in TAPE1 temporarily while the other is

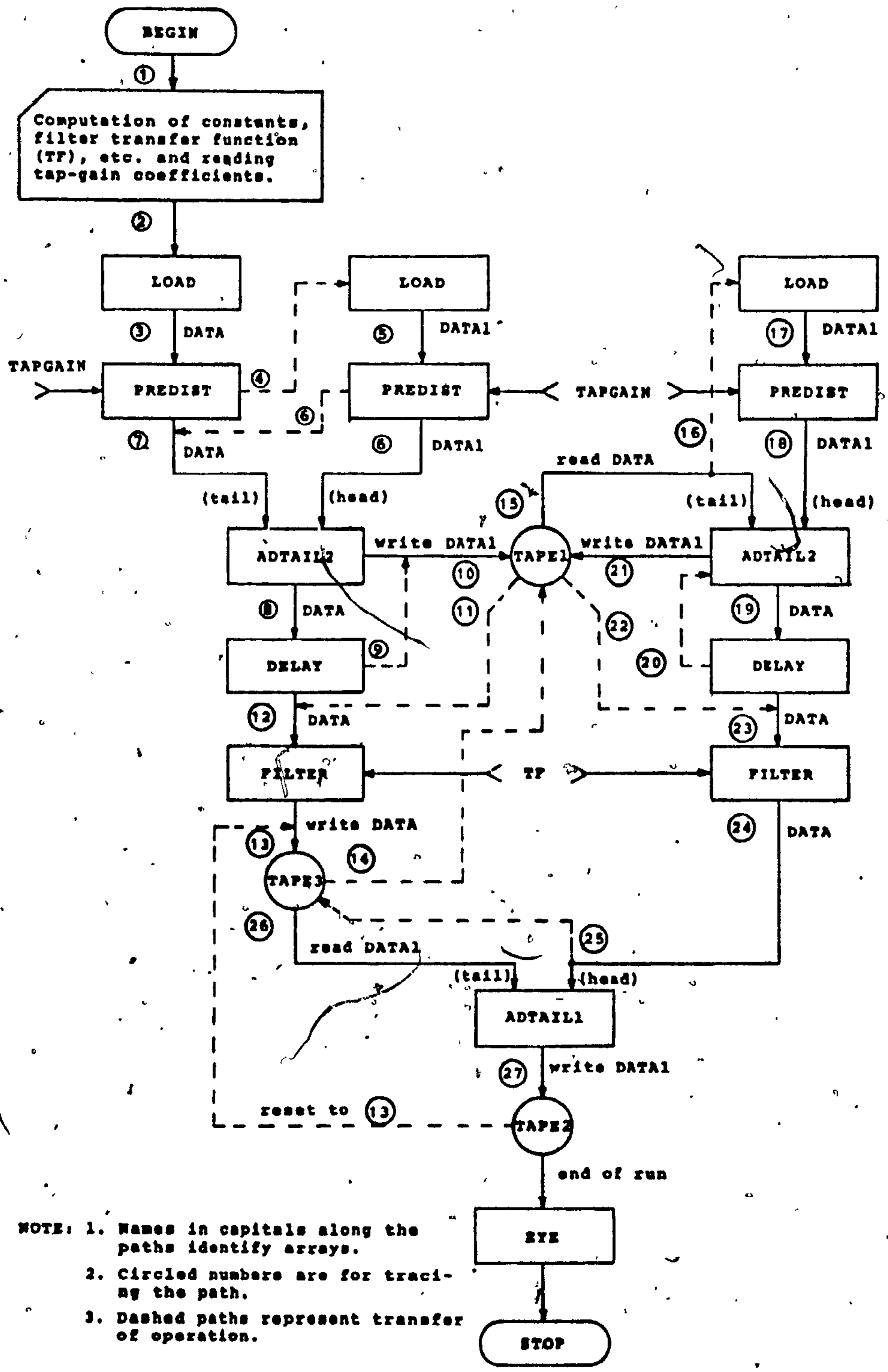


Fig.C.6 Flow Chart for SIMULA.

compensated for the delay caused by the predistortion and is, then, filtered. The filtered section is stored in TAPE3 temporarily while a third section is loaded and processed until the tail-head addition and delay compensation after predistortion. The second section is now filtered and again the tail-head adding operation is performed with the first section recalled from TAPE3. At this stage, the first section of the signal is processed completely. Once a section is processed completely it is stored permanently in TAPE2 followed by the next completely processed sections. When all loops of the run are completed (the number of loops being established at the beginning of the run), TAPE2 is rewound by subroutine EYE and the eye diagram of the filtered signal is printed. The listing of the program further describes each step of the simulation through its comment cards.

```
74/835   CPT=1                              FTN 4.6+528          31/08/10. 19.31.28        PAGE     1

      PROGRAM SIMULA (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3)
*****************************************************************
*  THIS PROGRAM SIMULATES THE ACTION OF PREDISTORTING AND FILTERING  *
*  PROCESSES OF A BASEBAND NRZ RANDOM SIGNAL.  FAST FOURIER TRANSF-  *
*  ORMATION (FFT) AND INVERSE FFT ARE USED TO CONVERT THE SIGNAL     *
*  FROM TIME-DOMAIN TO FREQUENCY-DOMAIN AND VICE VERSA.  FFT IS DONE *
*  THROUGH SUBROUTINES FROM THE IMS LIBRARY.  ACCESS TO IMSLIB MUST  *
*  BE MADE BEFORE RUNNING THIS PROGRAM.                              *
*                                                                    *
*  THE PARAMETER "ICODE" SPECIFIES THE TYPE OF LPF USED IN SIMULATION. *
*   ICODE = 0, THEORETICAL CHEBYCHEV OR BUTTERWORTH LPF.            *
*              (POLYNOMIAL COEFFICIENTS AND FILTER ORDER MUST BE GIVEN) *
*           1, PRACTICAL FILTER.(POLYN. COEFF. MUST BE GIVEN)       *
*           2, RAISED COSINE FILTER WITH (X/SIN(X)) MODIFICATION.   *
*              (EXCESS NYQUIST BANDWIDTH(ALFA) MUST BE GIVEN.)      *
*           3, NYQUIST FILTER. (ALFA MUST BE SPECIFIED)             *
*           4, IDEAL FILTER (BRICK-WALL TYPE).                      *
*                                                                    *
*  BY DEFAULT THE RESULTS PRINTED OUT ARE THE INPUT DATA, FILTER     *
*  TRANSFER FUNCTION AND THE EYE DIAGRAM OF THE SIGNAL AS SEEN       *
*  AT THE (TRANSMIT) FILTER OUTPUT. INTERMEDIATE RESULTS MAY, HOWEVER, *
*  BE OBTAINED THROUGH THE SWITCHES "SW" AS SHOWN BELOW.             *
*                                                                    *
*   SETTING "SW(J) = .TRUE." PRODUCES THE FOLLOWING PRINTOUTS -      *
*                                                                    *
*      SW(1) = THE SECTION OF SIGNAL ARRAY LOADED AS INPUT           *
*      SW(2) = PREDISTORTED SIGNAL ARRAY                             *
*      SW(3) = PREDISTORTED SIGNAL ARRAY AFTER TAIL-HEAD ADDITION    *
*      SW(4) = PREDISTORTED SIGNAL ARRAY AFTER DELAY COMPENSATION    *
*      SW(5) = SIGNAL ARRAY AFTER FILTERING AND ALSO THE FILTERED    *
*              SIGNAL AFTER TAIL-HEAD ADDITION                       *
*                                                                    *
*   THE NUMBER OF TAPS IN TRANSVERSAL FILTER MUST AT LEAST BE 5 AND AT *
*   MOST BE 27.  THE PREDISTORTION ACTION MAY BE SKIPPED BY LETTING  *
*   N=5, TAPGAIN(3)=1.0 AND ALL OTHER TAPGAINS=0.                    *
*****************************************************************
      DIMENSION DATA(1250),TF(1250),DATA1(1250),NX(11),TAPGAIN(33)
      LOGICAL SW(5)
      REWIND 2
      DATA JLAST,JTAP,M,LSAMPL/7,1,9,16/
C
C  THE FOLLOWING CONSTANTS MUST BE INITIALIZED WITH ZERO -
C          LOOP,II.
C
      DATA LOOP,II/2*0/
      READ *,LLTNOM,ICODE,FBAND,BIRATE,SW
      LDIM=K1=2**M
      N=LDIM/2
C
C  FILTER BANDWIDTH MUST BE MADE DOUBLE-SIDED TO MAKE "B.T=1."
C
      FBANDW=FBAND*2.
C
C  SET INITIAL SHIFT REGISTER CONTENTS.
C  THE PN-SEQUENCE IS OBTAINED BY SETTING ALL N(I)'S TO -1.
C
      DO 11 I=1,JLAST
```

```
   11    NX(I)=-1
C
C  CALCULATE THE NECESSARY CONSTANTS FOR VARIOUS SUBROUTINES.
C
      CALL CALCON(LSAMPL,LLTIOM,LLT,KKK,NSYMB,JLAST,LOCPM,LDIM,BIRATE,
     +SBANDW)
C
C  GENERATE TRANSMITTER FILTER TRANSFER FUNCTION.
C
      IF(ICODE.EQ.0) CALL THEOFIL(SBANDW,FBANDW,TF,LDIM)
      IF(ICODE.EQ.1) CALL MEASFIL(SBANDW,FBANDW,TF,LDIM)
      IF(ICODE.EQ.2) CALL RCCSMOD (SBANDW,BIRATE,TF,LDIM)
      IF(ICODE.EQ.3) CALL SCLTF6 (SBANDW,BIRATE,TF,LDIM)
      IF(ICODE.EQ.4) CALL IUFIL(SBANDW,FBANDW,TF,LDIM,0)
      PRINT 1,ICODE,FBAND,BIRATE,LSAMPL,SBANDW,LLTIOM,JLAST,KKK,
     +LLT,NSYMB,LOCPM
      PRINT 3,(TF(I),I=1,LDIM)
C
C  DETERMINE CORRECT STARTING SAMPLING POINT OF THE SIGNAL ARRAY
C  WHEN FILTERED.
C
      CALL BISINC(DATA,TF,LSAMPL,NSTART,LDIM,N,M)
      J=NSTART+2*KKK*LSAMPL-1
C
C  READ TAP-GAIN COEFFICIENTS FOR PREDISTORTION.
C  THE VALUE OF NN MUST, AT LEAST, BE 5.
C
      READ *,NN,(TAPGAIN(I),I=1,NN)
      PRINT 4,NN
      PRINT 12,(I,TAPGAIN(I),I=1,NN)
      LDIM=K2=2*LSAMPL*(KKK+NN-1)
C
C  LOAD FIRST SECTION OF SIGNAL INTO ARRAY DATA.
C
      CALL LOAD(DATA,KKK,LSAMPL,NX,II,JLAST,JTAP,LDIM)
      IF(SW(1)) 10,20
   10 PRINT 2
      PRINT 3,(DATA(I),I=1,LDIM)
   20 N2=LDIM-2*LSAMPL*(NN-1)
C
C  PREDISTORT THE SIGNAL ARRAY DATA.
C
      CALL PREDIST (DATA,LDIM,LSAMPL,NN,TAPGAIN,KKK)
      IF(SW(2)) 30,40
   30 PRINT 5
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  LOAD THE SECOND SECTION OF SIGNAL INTO ARRAY DATA1.
C
   40 CALL LOAD(DATA1,KKK,LSAMPL,NX,II,JLAST,JTAP,LDIM)
      IF(SW(1)) 50,60
   50 PRINT 2
      PRINT 3,(DATA1(I),I=1,LDIM)
C
C  PREDISTORT THE SIGNAL IN ARRAY DATA1.
C
   60 CALL PREDIST(DATA1,LDIM,LSAMPL,NN,TAPGAIN,KKK)
```

```
      IF(SW(2)) 70,80
   70 PRINT 5
      PRINT 3,(DATA1(I),I=1,LDIM)
C
C  ADD THE TAIL PORTION OF FIRST SECTION (DATA) OF PREDISTORTED
C  SIGNAL TO THE HEAD PORTION OF SECOND SECTION (DATA1) OF
C  PREDISTORTED SIGNAL TO OBTAIN A CONTINUOUS STREAM OF
C  PREDISTORTED SIGNAL.
C
   80 CALL ADTAIL2(DATA,DATA1,N2,LDIM)
      IF(SW(3)) 90,100
   90 PRINT 6
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  STORE THE CURRENT ARRAY DATA1 IN TAPE1.
C
  100 REWIND 1
      WRITE(1) DATA1
C
C  COMPENSATE FOR THE DELAY CAUSED BY PREDISTORTION.
C
      CALL DELAY(DATA,N1,LSAMPL,LDIM,KKK)
      IF(SW(4)) 110,120
  110 PRINT 7
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  PASS THE FIRST SECTION (DATA) OF PREDISTORTED SIGNAL
C  THROUGH THE TRANSMIT FILTER.
C
  120 LDIM=K1
      CALL FILTER(DATA,TF,N,M,LDIM)
      IF(SW(5)) 130,140
  130 PRINT 8
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  STORE THE FILTERED SECTION (DATA) IN TAPE3.
C
  140 REWIND 3
      WRITE(3) DATA
C
C  RECALL PREVIOUS SECTION OF PREDISTORTED SIGNAL AT THE
C  INPUT OF TRANSMIT FILTER FROM TAPE1 INTO ARRAY DATA.
C
      LDIM=K2
      REWIND 1
      READ(1) DATA
C
C  LOAD THE NEXT SECTION OF SIGNAL INTO ARRAY DATA1.
C
      CALL LOAD(DATA1,KKK,LSAMPL,NX,II,JLAST,JTAP,LDIM)
      IF(SW(1)) 150,160
  150 PRINT 2
      PRINT 3,(DATA1(I),I=1,LDIM)
C
C  PREDISTORT THE CURRENT SECTION (DATA1) OF SIGNAL.
C
  160 CALL PREDIST(DATA1,LDIM,LSAMPL,NN,TAPGAIN,KKK)
```

```
      IF(SW(2)) 170,180
  170 PRINT 5
      PRINT 3,(DATA1(I),I=1,LDIM)
C
C  ADD THE TAIL PORTION OF PREDISTORTED SECTION (DATA) OF
C  SIGNAL TO THE HEAD PORTION OF PREDISTORTED SECTION (DATA1)
C  OF SIGNAL TO OBTAIN A CONTINUOUS STREAM OF PREDISTORTED
C  SIGNAL.
C
  180 CALL ADTAIL2(DATA,DATA1,N2,LDIM)
      IF(SW(3)) 190,200
  190 PRINT 6
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  STORE THIS SECTION (DATA1) OF PREDISTORTED SIGNAL IN TAPE1.
C
  200 REWIND 1
      WRITE(1) DATA1
C
C  COMPENSATE FOR THE DELAY CAUSED BY PREDISTORTION.
C
      CALL DELAY(DATA,NN,LSAMPL,LDIM,KKK)
      IF(SW(4)) 210,220
  210 PRINT 7
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  PASS THE PRESENT SECTION (DATA) OF PREDISTORTED SIGNAL
C  THROUGH THE TRANSMIT FILTER.
C
  220 LDIM=K1
      CALL FILTER(DATA,TF,N,M,LDIM)
      IF(SW(5)) 230,240
  230 PRINT 8
      PRINT 3,(DATA(I),I=1,LDIM)
C
C  RECALL PREVIOUS SECTION OF FILTERED SIGNAL AFTER TRANSMIT
C  FILTER FROM TAPE3 INTO ARRAY DATA1.
C
  240 REWIND 3
      READ(3) DATA1
C
C  ADD THE TAIL PORTION OF FILTERED SECTION (DATA)  OF
C  SIGNAL TO THE HEAD PORTION OF THE FILTERED SECTION (DATA1)
C  OF SIGNAL TO OBTAIN A CONTINUOUS STREAM OF FILTERED SIGNAL.
C
      CALL ADTAIL1(DATA1,DATA,LLT,LDIM)
      IF(SW(5)) 250,260
  250 PRINT 9,LOOP
      PRINT 3,(DATA1(I),I=1,LDIM)
  260 IF(LOOP.EQ.0) GO TO 270
      DO 33 I=NSTART,J,2
   33    WRITE(2) DATA1(I)
  270 LOOP=LOOP+1
      IF(LOOP.LE.LOOPM) GO TO 140
      CALL EYE(LSAMPL,NSYMB)
    1 FORMAT(1H0//5X,*INPUT DATA AND COMPUTED PARAMETERS.*//,5X,
     +*TYPE OF FILTER USED IN SIMULATION, ICODE*,T55,*= *,I5/5X,
```

```
PROGRAM SIMULA      7-7635  OPT=1                         FTN 4.5+528        81/08/10. 19.31.28        PAGE    5

             +*FILTER BANDWIDTH*,T55,*= *,F10.4,* KHZ*/5X,*BIT RATE*,T55,
             +*= *,F10.4,* KBAUDS*/5X,*NO. SAMPLES PER BAUD*,T55,*= *,I5,/5X,
             +*SIMULATION BANDWIDTH*,T55,*= *,F10.4,* KHZ*/5X,
             +*APPROX. NO. LOCATIONS RESERVED FOR SPILL-OVER*,T55,*= *,I5/5X,
             +*SHIFT-REGISTER LENGTH *,T55,*= *,I5/5X,
             +*NO. SYMBOLS PROCESSED IN A LOOP*,T55,*= *,I5/5X,
             +*EXACT NO. LOCATIONS RESERVED FOR SPILL-OVER*,T55,*= *,I5/5X,
             +*TOTAL NO. SYMBOLS PROCESSED*,T55,*= *,I5/5X,
             +*MAX. NO. LOOPS NEEDED*,T55,*= *,I5///5X,
             +*FILTER TRANSFER FUNCTION.*//)
            2 FORMAT(//5X,*SIGNAL LOADED AS INPUT.*//)
            3 FORMAT(1X(2X,F8.3))
            4 FORMAT(//5X,I3,* TAP-GAIN COEFFICIENTS FOR PREDISTORTION.*//)
            5 FORMAT(//5X,*PREDISTORTED SIGNAL ARRAY.*//)
            6 FORMAT(//5X,*PREDISTORTED SIGNAL AFTER TAIL-HEAD ADDITION.*//)
            7 FORMAT(//5X,*DELAY COMPENSATED PREDISTORTED SIGNAL.*//)
            8 FORMAT(//5X,*FILTERED SIGNAL ARRAY.*//)
            9 FORMAT(//5X,*FILTERED SIGNAL AFTER TAIL-HEAD ADDITION IN*,
             +* LOOP *,I3,*.*//)
           12 FORMAT(4(10X,I2,3H.) ,E15.8)/)
              STOP
              END


      SYMBOLIC REFERENCE MAP (R=1)

 ENTRY POINTS
 12346   SIMULA

 VARIABLES     SN  TYPE              RELOCATION
 13527  BIRATE     REAL                               13547  DATA       REAL        ARRAY
 20453  DATA1      REAL         ARRAY                 13526  FBAND      REAL
 13533  FBANDW     REAL                               13534  I          INTEGER
 13525  ICODE      INTEGER                            13076  II         INTEGER
 13543  J          INTEGER                            13071  JLAST      INTEGER
 13072  JTAP       INTEGER                            13536  KKK        INTEGER
 13531  K1         INTEGER                            13545  K2         INTEGER
 13538  LDIM       INTEGER                            13535  LLT        INTEGER
 13524  LLINDM     INTEGER                            13075  LOOP       INTEGER
 13540  LOOPM      INTEGER                            13074  LSAMPL     INTEGER
 13073  M          INTEGER                            13532  N          INTEGER
 13544  NN         INTEGER                            13542  NSTART     INTEGER
 13537  NSYMB      INTEGER                            23015  NX         INTEGER     ARRAY
 13546  N2         INTEGER                            13541  SBANDW     REAL
 23371  SW         LOGICAL      ARRAY                 23030  TAPGAIN    REAL        ARRAY
 16111  TF         REAL         ARRAY

 FILE NAMES        MODE
     0  INPUT      FREE        2054  OUTPUT    FMT          4130  TAPE1     UNFMT        6204  TAPE2    UNFMT
 10260  TAPE3      UNFMT

 EXTERNALS           TYPE    ARGS
        ADTAIL1               4                        ADTAIL2               4
        BISINC                7                        CALCON               10
```

C.3.b Description of NOISGEN

The method of processing, here, is the same as that in SIMULA except that no predistortion is performed on the signal. Instead of processing three consecutive sections of signal as in SIMULA, only two sections need be processed simultaneously. When a section is processed completely it is stored permanently in TAPE2. When all the sections are processed, the filterd signal is read from TAPE2 to print a histogram of its samples depicting its amplitude density function that is superimposed with a theoretical gaussian probability density function having the same mean and standard deviation. The samples are also analyzed for statistical evidence of gaussian character. A flow chart for the program is shown in Fig.C.7. The listing of NOISGEN further describes the program through the comment cards.

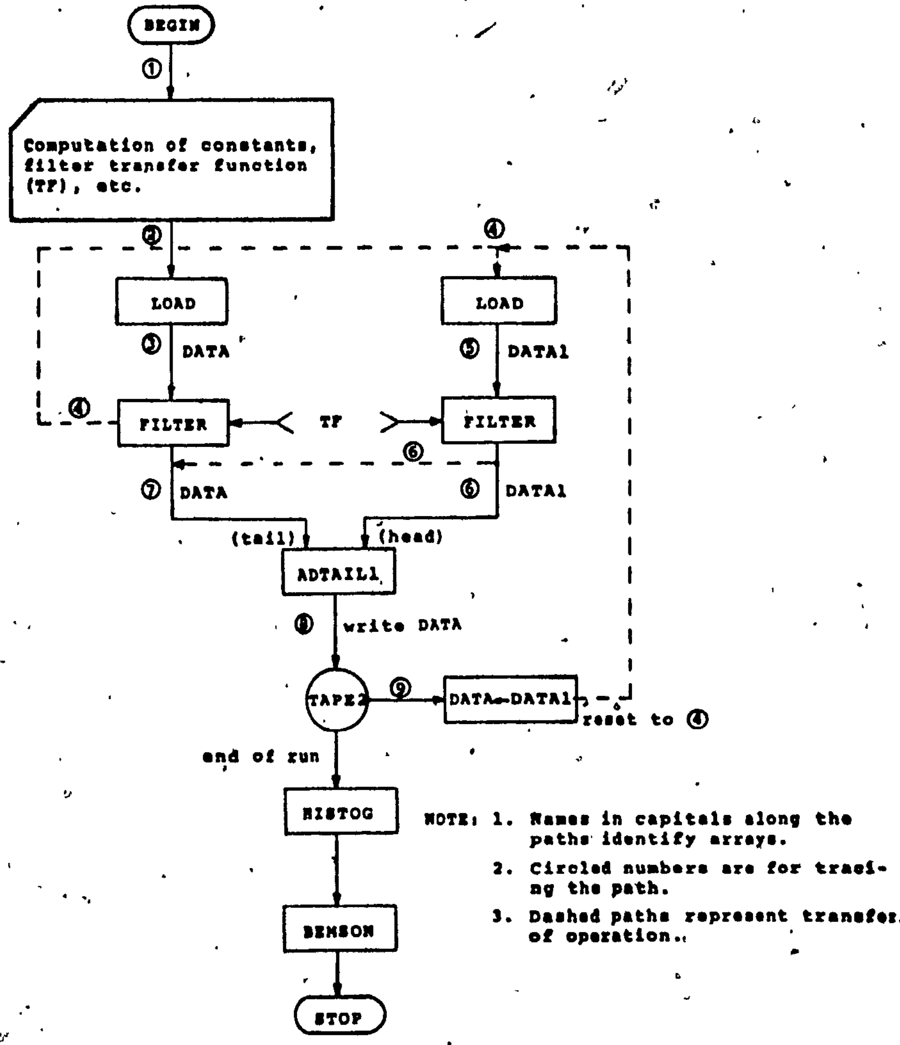


Fig.C.7 Flow Chart for NOISGEN.

```
      PROGRAM NOISGEN (INPUT,OUTPUT,TAPE2)
*********************************************************************
*  THIS PROGRAM SIMULATES PSEUDO-RANDOM NOISE BY FILTERING A PSEUDO-  *
*  RANDOM NRZ SIGNAL STREAM. FAST FOURIER TRANSFORMATION IS USED TO   *
*  CONVERT THE SIGNAL FROM TIME-DOMAIN INTO FREQUENCY-DOMAIN PRIOR TO *
*  FILTERING. AFTER MULTIPLICATION BY FILTER TRANSFER FUNCTION THE    *
*  SIGNAL IS BROUGHT BACK INTO TIME-DOMAIN THROUGH INVERSE FFT.  SUB- *
*  ROUTINES FOR FFT AND HISTOGRAM ARE ACCESSED FROM IMS LIBRARY.      *
*                                                                     *
*  BY DEFAULT THE RESULTS PRINTED OUT ARE THE INPUT DATA, FILTER      *
*  TRANSFER FUNCTION AND THE HISTOGRAM OF SIGNAL PROBABILITY DENSITY  *
*  FUNCTION AT FILTER OUTPUT. INTERMEDIATE RESULTS, HOWEVER, MAY BE   *
*  OBTAINED THROUGH SWITCHES 'SW' AS SHOWN BELOW.                     *
*                                                                     *
*   "SETTING SW(J) = .TRUE." PRODUCES THE FOLLOWING PRINTOUTS -       *
*                                                                     *
*     SW(1) = THE SECTION OF SIGNAL ARRAY LOADED AS INPUT             *
*     SW(2) = SIGNAL ARRAY AFTER FILTERING AND ALSO THE FILTERED      *
*                  SIGNAL AFTER TAIL-HEAD ADDITION                    *
*********************************************************************
      DIMENSION DATA(1024),TF(1024),DATA1(1024),NX(11)
      LOGICAL SW(2)
      REWIND 2
      DATA LSAMPL/16/
C
C  THE FOLLOWING CONSTANTS MUST BE INITIALIZED WITH ZERO -
C           LOOP,IT.
C
      DATA LOOP,IT/2*0/
      READ *,JLAST,JTAP,M,LLTNOM,FBAND,BIRATE,SW
      LDIM=2**M
      N=LDIM/2
C
C  FILTER BANDWIDTH MUST BE MADE DOUBLE-SIDED TO MAKE "B.T=1."
C
      FBANDW=FBAND*2.
C
C  SET INITIAL SHIFT REGISTER CONTENTS.
C  THE PN-SEQUENCE IS OBTAINED BY SETTING ALL NX(I)'S TO -1.
C
      DO 11 I=1,JLAST
   11 NX(I)=-1
C
C  CALCULATE THE NECESSARY CONSTANTS FOR VARIOUS SUBROUTINES.
C
      CALL CALCON(LSAMPL,LLTNOM,LLT,KKK,NSYMB,JLAST,LOOPM,LDIM,BIRATE,
     +SBANDW)
C
C  GENERATE FILTER TRANSFER FUNCTION.
C
      CALL THEOFIL(SBANDW,FBANDW,TF,LDIM)
      PRINT 1,FBAND,BIRATE,LSAMPL,SBANDW,LLTNOM,JLAST,KKK,
     +LLT,NSYMB,LOOPM,SW
C
C  DETERMINE CORRECT STARTING SAMPLING POINT OF THE SIGNAL
C  ARRAY AFTER FILTERING.
C
```

CONCORDIA UNIVERSITY COMPUTER CENTRE

```
PROGRAM NOISGEN     74/635  OPT=1                          FTN 4.3+528        81/03/11. 17.33.56        PAGE    2

          CALL BISINC(DATA,TF,LSAMPL,NSTART,LDIM,N,M)
          J=NSTART+2*KKK*LSAMPL-1
C
C  LOAD FIRST SECTION OF SIGNAL INTO ARRAY DATA.
C
          CALL LOAD(DATA,KKK,LSAMPL,NX,II,JLAST,JTAP,LDIM)
          IF(SW(2)) 10,20
       10 PRINT 2
          PRINT 3,(DATA(I),I=1,LDIM)
C
C  PASS DATA THROUGH THE FILTER.
C
       20 CALL FILTER(DATA,TF,N,M,LDIM)
          IF(SW(2)) 30,40
       30 PRINT 4
          PRINT 3,(DATA(I),I=1,LDIM)
C
C  LOAD NEXT SECTION OF SIGNAL ARRAY INTO DATA1.
C
       40 CALL LOAD(DATA1,KKK,LSAMPL,NX,II,JLAST,JTAP,LDIM)
          IF(SW(2)) 50,60
       50 PRINT 2
          PRINT 3,(DATA1(I),I=1,LDIM)
C
C  PASS DATA1 THROUGH THE FILTER.
C
       60 CALL FILTER(DATA1,TF,N,M,LDIM)
          IF(SW(2)) 70,80
       70 PRINT 4
          PRINT 3,(DATA1(I),I=1,LDIM)
C
C  ADD THE TAIL PORTION OF FILTERED SECTION "DATA" OF
C  SIGNAL TO THE HEAD PORTION OF THE FILTERED SECTION "DATA1"
C  OF SIGNAL TO OBTAIN A CONTINUOUS STREAM OF FILTERED SIGNAL.
C
       80 CALL ADTAIL1(DATA,DATA1,LLT,LDIM)
          IF(SW(2)) 90,100
       90 PRINT 5,LOOP
          PRINT 3,(DATA(I),I=1,LDIM)
      100 IF(LOOP.EQ.0) GO TO 110
          DO 33 I=NSTART,J,2
       33    WRITE(2) DATA(I)
      110 LOOP=LOOP+1
          DO 44 I=1,LDIM
       44    DATA(I)=DATA1(I)
          IF(LOOP.LE.LOOPM) GO TO 40
          CALL HISTOG
        1 FORMAT(1H0//5X,*INPUT DATA AND COMPUTED PARAMETERS.*//,5X,
         +*FILTER BANDWIDTH*,T55,*= *,F10.4,* KHZ*/5X,*BIT RATE*,T55,
         +*= *,F10.4,* KBAUDS*/5X,*NO. SAMPLES PER BAUD*,T55,*= *,I5/5X,
         +*SIMULATION BANDWIDTH*,T55,*= *,F10.4,* KHZ*/5X,
         +*APPROX. NO. LOCATIONS RESERVED FOR SPILL-OVER*,T55,*= *,I5/5X,
         +*SHIFT-REGISTER LENGTH *,T55,*= *,I5/5X,
         +*NO. SYMBOLS PROCESSED IN A LOOP*,T55,*= *,I5/5X,
         +*EXACT NO. LOCATIONS RESERVED FOR SPILL-OVER*,T55,*= *,I5/5X,
         +*TOTAL NO. SYMBOLS PROCESSED*,T55,*= *,I5/5X,
         +*MAX. NO. LOOPS NEEDED*,T55,*= *,I5/5X,
```

```
PROGRAM NOISGEN    74/835  OPT=1                              FTN 4.6+528        31/03/11. 17.30.56        PAGE    3

             '**SWITCHES SW1 & SW2 FOR PRINTOUTS*,T55,**  *,2L2//)
             2 FORMAT(//5X,*SIGNAL LOADED AS INPUT.*//)
             3 FORMAT(15(2X,F5.3))
             4 FORMAT(//5X,*FILTERED SIGNAL ARRAY.*//)
             5 FORMAT(//5X,*FILTERED SIGNAL AFTER TAIL-HEAD ADDITION.*,
              +*    LOOP =*,I3//)
               STOP
               END

     SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
  6214  NOISGEN

VARIABLES       SN  TYPE           RELOCATION
  6705  BIRATE      REAL                               6721  DATA     REAL       ARRAY
 12721  DATA1       REAL        ARRAY                  5704  FRAND    REAL
  6710  FRANDW      REAL                               5711  I        INTEGER
  5455  II          INTEGER                            5723  J        INTEGER
  5700  JLAST       INTEGER                            5701  JTAP     INTEGER
  6713  KKK         INTEGER                            5706  LDIM     INTEGER
  6722  LLT         INTEGER                            5703  LLTNUM   INTEGER
  6454  LOOP        INTEGER                            6715  LOOPM    INTEGER
  6453  LSAMPL      INTEGER                            5702  M        INTEGER
  5707  N           INTEGER                            5717  NSTART   INTEGER
  5714  NSYMB       INTEGER                           14721  NX       INTEGER    ARRAY
  5716  SRANDW      REAL                              14734  SW       LOGICAL    ARRAY
 10721  TF          REAL        ARRAY

FILE NAMES      MODE
   0  INPUT     FREE        2056  OUTPUT    FMT          4130  TAPE2     UNFMT

EXTERNALS            TYPE   ARGS
        ADTAIL1              4                          BISINC          7
        CALCON              10                          FILTER          5
        HISTOG               0                          LOAD            4
        THEOFIL              4

STATEMENT LABELS
  6555   1      FMT                     5531   2      FMT                        5555   3       FMT
  6663   4      FMT                     5655   5      FMT                           0  10         INACTIVE
     0  11                              5251  20                                    0  30         INACTIVE
     0  33                              5273  40                                    0  44
     0  50         INACTIVE             5305  60                                    0  70         INACTIVE
  6317  80                                 5  90         INACTIVE                6931 100
  6343 110

LOOPS  LABEL    INDEX     FROM-TO    LENGTH    PROPERTIES
  5231  11      I          41 42       28        OPT
  6334  33      I          98 99       78                   EXT REFS
  6347  44      I         101 102      38        OPT
```

C.3.c Description of PULSE

This is an extraction from NOISGEN, but modified to simulate the response of a filter excited by an isolated pulse. Instead of loading the array DATA with KKK symbols as in previous programs, a single pulse sampled in time-domain 16 times, is converted into frequency-domain, multiplied by the filter (theoretical or practical) transfer function and converted back to time-domain. The listing of pulse is given next.

PROGRAM PULSE 74/835 OPT=1 FTN 4.6+528 81/08/16. 17.23.08 PAGE

```
      PROGRAM PULSE(INPUT,OUTPUT,TAPE1)
***************************************************************
*   THIS PROGRAM COMPUTES THE SAMPLES OF THE RESPONSE OF A   *
*   GIVEN FILTER (SW=.T. MEANS THEORETICAL FILTER AND .F.    *
*   MEANS PRACTICAL FILTER) TO AN INPUT PULSE. THE NUMBER    *
*   OF SAMPLES CORRESPONDS TO THE SPECIFIED LSAMPL.          *
***************************************************************
      DIMENSION DATA(1024),TF(1024)
      LOGICAL SW
      REWIND 1
C
C   THE CONSTANTS LOOP, II MUST BE INITIALIZED WITH ZERO.
C
      DATA LOOP,II/2*0/
      READ *,M,LSAMPL,FBAND,BIRATE,SW
      LDIM=2**M
      N=LDIM/2
C
C   FILTER BANDWIDTH MUST BE MADE DOUBLE-SIDED TO MAKE "B.T=1."
C
      FBANDW=2.*FBAND
C
C     CALCULATE NECESSARY CONSTANTS FOR VARIOUS SUBROUTINES.
C
      CALL CALCON(LSAMPL,0,LLT,KKK,NSYMB,JLAST,LCOPM,LDIM,BIRATE,
     +SBANDW)
C
C   GENERATE FILTER TRANSFER FUNCTION.
C
      IF(SW) CALL THEOFIL(SBANDW,FBANDW,TF,LDIM)
      IF(.NOT.SW) CALL MEASFIL(SBANDW,FBANDW,TF,LDIM)
      PRINT 1,FBAND,BIRATE,LSAMPL,SBANDW
      PRINT 2,(TF(I),I=1,LDIM)
C
C   DETERMINE THE CORRECT STARTING SAMPLING POINT (NSTART)
C   OF SIGNAL AT FILTER OUTPUT.
C
      CALL BISINC(DATA,TF,LSAMPL,NSTART,LDIM,N,M)
C
C   INITIALIZE THE ARRAY DATA(.). PUT 1'S TO THE FIRST 16 REAL
C   POINTS AND 0'S TO THE REST AND ALL IMAGINARY POINTS OF THE ARRAY.
C   THIS SIMULATES A SINGLE ISOLATED PULSE THAT IS SAMPLED 16 TIMES.
C
      DO 11 I=1,LDIM
   11 DATA(I)=0.
      LSAMP2=2*LSAMPL
      II=LSAMP2-1
      DO 22 I=1,II,2
   22 DATA(I)=1.
      PRINT 6
      PRINT 2,(DATA(I),I=1,LDIM)
C
C     PASS THE SIGNAL THROUGH THE FILTER.
C
      CALL FILTER(DATA,TF,N,M,LDIM)
      PRINT 3
      PRINT 2, (DATA(I),I=1,LDIM)
```

CONCORDIA UNIVERSITY COMPUTER CENTRE

```
PROGRAM PULSE       74/835  OPT=1                               FTN 4.6+528       81/08/16. 17.23.08      PAGE    2

                  II=NSTART+LSAMPL
               10 IF(II.LE.LSAMP2) GO TO 20
                  II=II-LSAMP2
                  GO TO 10
               20 PRINT 5
                  PRINT *,(DATA(I),I=II,LDIM,LSAMP2)
                  DO 33 I=II,LDIM,LSAMP2
               33    WRITE(1,*) DATA(I)
                1 FORMAT(1H0,4X,*INPUT DATA AND COMPUTED PARAMETERS.*//5X,
                 +*FILTER BANDWIDTH*,T45,*=*,F10.4,* KHZ*/5X,*BIT RATE*,T45,
                 +*=*,F10.4,* KBAUDS*/5X,*NO. SAMPLES PER BAUD*,T45,*=*,I5/5X,
                 +*SIMULATION BANDWIDTH*,T45,*=*,F10.4,* KHZ*//5X,
                 +*FILTER TRANSFER FUNCTION.*//)
                2 FORMAT(16(2X,F6.3))
                3 FORMAT(1H0//5X,*PULSE RESPONSE OF THE FILTER.*//)
                5 FORMAT(1H0//5X,*SAMPLED PULSE RESPONSE OF FILTER.*//)
                6 FORMAT(1H0//5X,*INPUT DATA TO THE FILTER.*//)
                  STOP
                  END


         SYMBOLIC REFERENCE MAP (R=1)

 ENTRY POINTS
  6214   PULSE

 VARIABLES      SN  TYPE           RELOCATION
  6546   BIRATE      REAL                             6563   DATA       REAL         ARRAY
  6545   FBAND       REAL                             6551   FBANDW     REAL
  6568   I           INTEGER                          6403   II         INTEGER
  6555   JLAST       INTEGER                          6553   KKK        INTEGER
  6547   LDIM        INTEGER                          6552   LLT        INTEGER
  6482   LOOP        INTEGER                          6556   LOOPM      INTEGER
  6544   LSAMPL      INTEGER                          6562   LSAMP2     INTEGER
  6543   M           INTEGER                          6550   N          INTEGER
  6561   NSTART      INTEGER                          6554   NSYMB      INTEGER
  6557   SBANDW      REAL                             6542   SW         LOGICAL
 10563   TF          REAL          ARRAY

 FILE NAMES      MODE
     0   INPUT       FREE          2054  OUTPUT     MIXED         4130  TAPE1       FREE

 EXTERNALS          TYPE   ARGS
         BISINC             7                        CALCOM           10
         FILTER             5                        MEASFIL           4
         THEOFIL            4

 STATEMENT LABELS
  6460   1      FMT                6513   2      FMT                6516   3      FMT
  6524   5      FMT                6532   6      FMT                6307   10
     0   11                        6313   20                           0   22
     0   33
```

C.3.d <u>Description of PEAKDIS</u>

The program assumes that the predistorter (transversal filter) has (2N+1) taps, symmetrical about the center tap and that the input pulse has (2M+1) samples with its peak represented by the $(M+1)^{th}$ sample. The program adjusts the number of samples to be (2M+2N+1), the resulting non-existent samples being assigned with zero amplitude. The listing is given next.

```
PROGRAM PEAKDIS     74/835   OPT=1                    FTN 4.6+528        81/08/05. 20.57.56        PAGE    1

      PROGRAM PEAKDIS(INPUT,OUTPUT)
C************************************************************************
C*    THIS PROGRAM COMPUTES THE OPTIMUM TAP-GAIN COEFFICIENTS FOR A    *
C*    (2N+1)-TAP TRANSVERSAL FILTER THAT IS DESIGNED AS A PREDISTORTING *
C*    DEVICE FOR A TRAIN OF *NRZ* PULSES, WHICH ARE PASSED THROUGH A   *
C*    BUTTERWORTH OR A CHEBYSCHEV LOWPASS FILTER. THE PROGRAM REQUIRES *
C*    (2M+1) SAMPLES OF PULSE RESPONSE OF THE LOWPASS FILTER. NUMBERS  *
C*    M & N MUST NOT EACH BE GREATER THAN 16. IF THEY DO, THE DIMENSION *
C*    OF ARRAYS MUST BE CHANGED AS FOLLOWS,.                           *
C*                XX ======>  (2M+2N+1)                                 *
C*                 A ======>  (2N+3),(2N+3)                             *
C*             B,E,S ======>  (2N+1)                                    *
C*                 X ======>  (2N+1)                                    *
C*                                                                      *
C*    TOLER = TOLERANCE IN COMPUTING THE LINEAR SYSTEM OF EQUATIONS.   *
C*    GAIN  = GAIN OF THE SUMMING DEVICE TO BE USED (OP. APP.)         *
C************************************************************************
      DIMENSION XX(66),A(35,35),B(33),X(33),E(33)
      INTEGER S(33)
      READ *,M,N,TOLER,GAIN
      LN=2*N+1
      LM=2*M+1
      LT=LN+LM-1
      PRINT 1,N,LT,TOLER
      DO 20 I=1,LM
         DO 10 J=1,LN
   10       A(I,J)=0.
   20    B(I)=XX(I)=XX(LM+I)=0.
      J=-M
      LP=LM+N-1
      DO 30 I=N,LM
         READ *,XX(I+1)
         PRINT 2,J,XX(I+1)
   30    J=J+1
      LM=2*M+1
      PRINT 3
      B(1)=XX(M+N+1)
      J=-M-N
      DO 40 I=1,LT
         XX(I)=XX(I)/B(1)
         PRINT 2,J,XX(I)
   40    J=J+1
      B(1)=0.
      M2=LM+2
      N2=LN+2
      CALL CAL(XX,A,B,X,E,S,LN,LM,LT,M2,N2,TOLER,GAIN)
    1 FORMAT(1H1,4X,*INPUT DATA*/5X,*----------*//
     +5X,*NO. TAPS IN THE EQUALIZER IS (2N+1), WHERE N*,T60,*=*,I3/
     +5X,*NO. PULSE SAMPLES REQUIRED FOR COMPUTATION*,T60,*=*,I3/
     +5X,*TOLERANCE IN COMPUTATION*,T60,*=*,E10.3//
     +5X,*INPUT PULSE SAMPLES GIVEN AS,*/)
    2 FORMAT(5X,*X(*,I3,*) = *,F10.6)
    3 FORMAT(/5X,*ALL OTHER SAMPLES ARE ASSUMED TO BE ZERO.*//
     +5X,*SAMPLES AFTER NORMALIZATION BY THE CENTER SAMPLE, X(0),*/)
      STOP
      END
```

C.3.e Description of CURVFIT

This program computes a polynomial up to 20^{th} degree to given set of data points (x- and y-axes). The maximum number of data points it accepts is only 50; if not, the program aborts. Optional plotting of the given data points and the fitted curve on the same axes is supplied by 'PLT= .TRUE.'. If statistical information with regard to the regression analysis is needed 'PNT=.TRUE.' must be supplied at input. Also if there is no improvement in the fitted curve below the specified degree, M (<20), the program terminates. To have this capability 'TERM= .TRUE.' must be given at input. The listing of CURVFIT is given next.

```
PROGRAM CURVFIT    74/435  OPT=1                    FTN 4.6+528      81/08/13. 13.11.19      PAGE    1

      PROGRAM CURVFIT (INPUT,OUTPUT,TAPE5,TAPE6)
*************************************************************
*   THIS PROGRAM COMPUTES  A  POLYNOMIAL EXPRESSION OF UP   *
*   TO 20TH DEGREE OF A CURVE THAT FITS A GIVEN SET OF      *
*   DATA VALUES (Y VS. X) UP TO 50 OBSERVATIONS.            *
*   IF VARIABLE "TERM" IS SPECIFIED ".TRUE." AT INPUT       *
*   THE EXECUTION TERMINATES WHEN THERE IS NO IMPROVEMENT   *
*   IN THE FIT FOR HIGHER DEGREE.                           *
*************************************************************
      REAL PRNAME(4),Y(50)
      REAL DI(1000),X(1050),D(400),E(30),P(150)
      DOUBLE PRECISION DI,Y,X,D,B,P
      LOGICAL PNT,PLT,TERM
      DO 10 J=1,1050
   10 X(J)=0.
      READ (5,1) N,M,PLT,PNT,TERM,PRNAME
    1 FORMAT (2I5,3L1,2X,4A10)
      IF (N.GT.50) GO TO 995
      DO 11 J=1,N
   11 READ(5,*) X(J),Y(J)
C
C  ESTABLISH MAX ALLOWABLE ORDER
C
      IF (M.LE.0.OR.M.GT.20) M=20
C
C  CONSTRUCT X MATRIX
C
      L=N*M
      DO 40 J=1,N
      K=J+L
   40 X(K)=Y(J)
      CALL POLRGW (PRNAME,N,M,PLT,PNT,TERM,X,DI,D,P,B)
      GO TO 999
  995 WRITE (6,996)
  996 FORMAT (//* ABORT - NUMBER OF SAMPLES EXCEEDS 50*)
  999 STOP
      END

  SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
10271  CURVFIT

VARIABLES     SN  TYPE          RELOCATION
22241  B          DOUBLE      ARRAY            20601  D      DOUBLE    ARRAY
10575  DI         DOUBLE      ARRAY            10420  J      INTEGER
10424  K          INTEGER                      10423  L      INTEGER
10422  M          INTEGER                      10421  N      INTEGER
22335  P          DOUBLE      ARRAY            10416  PLT    LOGICAL
10415  PNT        LOGICAL                      10425  PRNAME REAL      ARRAY
10417  TERM       LOGICAL                      14515  X      DOUBLE    ARRAY
10431  Y          DOUBLE      ARRAY
```

C.4 Listing of Subroutines

Each subroutine is described briefly at the beginning of its listing. Further description to some of the subroutines are given by Huang [1979] in Appendix C2 of his Ph.D. thesis.

```
      SUBROUTINE ADTAIL1(ARRAYA,ARRAYB,LLT,LDIM)
*****************************************************************
*  THIS SUBROUTINE ADDS THE LAST LLT COMPLEX VALUES OF ARRAYA  *
*  TO THE FIRST LLT COMPLEX VALUES OF ARRAYB, AND THE VALUES   *
*  ARE REPLACED BY THE SUM.                                     *
*****************************************************************
      DIMENSION ARRAYA(LDIM),ARRAYB(LDIM)
      I2=2*LLT
      DO 11 II=1,I2
         IJ=LDIM-I2+II
         ARRAYB(II)=ARRAYB(II)+ARRAYA(IJ)
   11    ARRAYA(IJ)=ARRAYB(II)
      RETURN
      END
```

```
      SUBROUTINE ADTAIL2(ARRAYA,ARRAYB,N2,LDIM)
*****************************************************************
*  THIS SUBROUTINE ADDS THE TAIL PORTION OF ARRAYA TO THE      *
*  HEAD PORTION OF ARRAYB ACCORDING TO THE VALUE ASSIGNED      *
*  TO PARAMETER N2. THESE PORTIONS ARE REPLACED BY THE SUM.    *
*  THIS SUBROUTINE IS NECESSARY WHEN PREDISTORTING THE         *
*  SIGNAL ARRAY.                                                *
*****************************************************************
      DIMENSION ARRAYA(LDIM),ARRAYB(LDIM)
      N1=LDIM-N2
      DO 11 I=1,N1
         J=N2+I
         ARRAYB(I)=ARRAYB(I)+ARRAYA(J)
   11    ARRAYA(J)=ARRAYB(I)
      RETURN
      END
```

```
      SUBROUTINE NISINC(DATA,TF,LSAMPL,NSTART,LDIM,N,M)
**************************************************************************
*  THIS SUBROUTINE DETERMINES THE START OF SAMPLING POSITION             *
*  OF A SIGNAL ARRAY AFTER IT PASSES THROUGH THE TRANSMITTER FILTER.     *
*                                                                        *
*  DATA   = ARRAY CONTAINING THE SIGNAL                                  *
*  TF     = ARRAY CONTAINING THE TRANSMITTER FILTER TRANSFER FUNCTION    *
*  LSAMPL = NO. OF SAMPLES PER SYMBOL                                    *
*  NSTART = BEFORE THE TRANSMITTER FILTERING THE FIRST SAMPLE OF        *
*           THE SIGNAL IS AT LOCATION 1, AFTER FILTERING THE STARTING    *
*           SAMPLE WILL BE AT NSTART.                                    *
*  LDIM   = DIMENSION OF ARRAYS DATA AND TF.                             *
**************************************************************************
      DIMENSION DATA(LDIM),TF(LDIM)
      INDEX=0
      DO 11 I=1,LDIM
   11    DATA(I)=0.
      LSAMP2=2*LSAMPL
      I1=LSAMP2-1
      DO 22 I=1,I1,2
   22    DATA(I)=1.
      CALL FILTER(DATA,TF,N,M,LDIM)
    1 SUM=0.
      DO 44 K=1,190,2
         I2=K+LSAMP2-2
         TSUM=0.
         DO 33 I=K,I2,2
   33       TSUM=TSUM+DATA(I)
         IF(TSUM.LT.SUM) GO TO 44
         SUM=TSUM
         KSTART=K
   44 CONTINUE
      INDEX=INDEX+1
      IF(INDEX.LE.1) GO TO 1
      NSTART=KSTART
      PRINT 2,NSTART
    2 FORMAT(//5X,*START OF SAMPLING AT FILTER OUTPUT, NSTART   = *,
     +I3//)
      RETURN
      END
```

```
      SUBROUTINE CAL(XX,A,B,X,E,S,LN,LM,LT,M2,N2,TOLER,GAIN)
C**************************************************************
C*   THIS SUBROUTINE COMPUTES THE NECESSARY CONSTANTS FOR    *
C*   CALLING THE SUBROUTINE L1 IN ORDER TO SOLVE THE OVER-   *
C*   DETERMINED SYSTEM OF LINEAR EQUATIONS.                  *
C**************************************************************
      DIMENSION XX(LT),A(M2,N2),B(LM),X(LN),E(LM)
      INTEGER S(LM)
      PRINT 1,LM,LN,LM
      DO 10 I=1,LM
         DO 10 J=1,LN
   10       A(I,J)=-XX(I-J+LN)
      J=LM/2
      B(J+1)=1.
      DO 20 I=1,LM
   20    PRINT 2,(A(I,J),J=1,LN),B(I)
      CALL L1(LM,LN,M2,N2,A,B,TOLER,X,E,S)
      I1=A(M2-1,N2)
      I2=A(M2,N2-1)
      I3=A(M2,N2)
      PRINT 3, A(M2-1,N2-1),I1,I2,I3
      IF(I2.NE.2) GO TO 11
      PRINT 4
      GO TO 99
   11 IF(I2.EQ.0) PRINT 5
      IF(I2.EQ.1) PRINT 6
      J=-LN/2
      PRINT 7
      DO 30 I=1,LN
         PRINT 8,J,X(I)
   30    J=J+1
      PRINT 14
      PRINT 13,(I,E(I),I=1,LM)
      B(1)=X(LN/2+1)
      PRINT 9
      J=-LN/2
      DO 40 I=1,LN
         X(I)=X(I)/B(1)
         B(I+1)=GAIN/X(I)
         PRINT 12,J,X(I),J,B(I+1)
   40    J=J+1
    1 FORMAT(1H1,4X,*MATRIX A(*,I2,*,*,I2,*) AND VECTOR B(*,I2,*),*//)
    2 FORMAT(13(2X,F8.5))
    3 FORMAT(1H1,4X,*RESULTS */5X,*********//5X,
     +*MINIMUM SUM OF ABSOLUTE VALUES OF THE RESIDUALS*,T60,
     +*= *,E16.10/5X,*THE RANK OF THE MATRIX *A**,T60,*= *,I3/
     +5X,*EXIT CODE OF THE SUBROUTINE*,T60,*= *,I3/
     +5X,*NUMBER OF ITERATIONS REQUIRED BY THE SIMPLEX METHOD*,
     +T60,*= *,I3/)
    4 FORMAT(//5X,*CALCULATIONS TERMINATED PREMATURELY DUE TO *,
     +*ROUNDING OFF ERRORS.*//)
    5 FORMAT(5X,*OPTIMAL SOLUTION, WHICH IS PROBABLY UNIQUE IS *
     +*FOUND.*//)
    6 FORMAT(5X,*UNIQUE OPTIMAL SOLUTION IS FOUND.*//)
    7 FORMAT(//5X,*SOLUTION TO THE PROBLEM.*/5X,
     +*TAP-GAIN COEFFICIENTS ARE,*//)
    8 FORMAT(5X,*C(*,I3,*) = *,E15.8)
    9 FORMAT(/5X,*COEFFICIENTS ARE NORMALIZED WITH RESPECT TO THE *
     +*CENTER COEFFICIENT, C(0).*//6X,*NORMALIZED COEFFICIENTS*,14X,
     +*  CORRESPONDING RESISTOR VALUE*/)
   12 FORMAT(5X,*C(*,I3,*) = *,E16.8,11X,*R(*,I3,*) = *,E16.8,
     +*  KILO-OHMS*)
   13 FORMAT(5X,*E(*,I2,*) = *,E15.8)
   14 FORMAT(//5X,*RESIDUALS IN THE EQUATIONS.*/)
   99 RETURN
      END
```

```
      SUBROUTINE CALCON(LSAMPL,LLINCH,LLT,KKK,NSYMB,JLAST,
     +LOOPN,LDIM,BIRATE,SBANDW)
C***********************************************************
C*  THIS SUBROUTINE CALCULATES SOME                          *
C*  CONSTANTS NEEDED IN THE OPERATION OF OTHER SUBROUTINES.  *
C*                                                           *
C*        LSAMPL = NO. SAMPLES PER SYMBOL.                   *
C*        LLINCH = APPROXIMATE NO. LOCATIONS RESERVED FOR    *
C*                 FILTER SPILLOVER EFFECT.                  *
C*        JLAST  = SHIFT REGISTER ARRAY LENGTH.              *
C*                                                           *
C*  THE FOLLOWING PARAMETERS ARE RETURNED.                   *
C*        KKK    = NO. SYMBOLS TO BE PROCESSED IN A LOOP.    *
C*        LLT    = EXACT NO. LOCATIONS RESERVED FOR FILTER   *
C*                 SPILLOVER EFFECT.                         *
C*        NSYMB  = TOTAL NO. SYMBOLS TO BE PROCESSED.        *
C*        LOOPN  = MAX. NO. CALCULATION LOOPS NEEDED.        *
C*        SBANDW = SIMULATION BANDWIDTH.                     *
C***********************************************************
      SBANDW=BIRATE*LSAMPL
      LL=LDIM/2
      AKK=(LL-LLINCH)/LSAMPL
      KKK=IFIX(AKK)
      LLT=LL-(KKK*LSAMPL)
      NSYMB=2**JLAST-1
      SYMB=NSYMB
      SKK=KKK
      POOL=SYMB/SKK
      LOOPN=IFIX(POOL)
      X=LOOPN
      IF(X.LT.POOL) LOOPN=LOOPN+1
      RETURN
      END
```

```
      SUBROUTINE DELAY(DATA,N,LSAMPL,LDIM,KKK)
C***********************************************************
C*  THIS SUBROUTINE COMPENSATES FOR THE DELAY CAUSED BY THE *
C*  PREDISTORTING PROCESS.                                   *
C***********************************************************
      DIMENSION DATA(LDIM)
      K=2*LSAMPL*(N/2)
      I1=2*LSAMPL*KKK
      DO 11 I=1,I1,2
         I2=I+K
         DATA(I)=DATA(I2)
   11    DATA(I+1)=DATA(I2+1)
      I1=I1+1
      DO 22 I=I1,LDIM
   22    DATA(I)=0.
      RETURN
      END
```

```
      SUBROUTINE EYE(NSB,MPS)
C*****************************************************************
C* THIS SUBROUTINE PRINTS A 131 X 61 MATRIX OF THE EYE DIAGRAM. *
C* THE DIMENSION 131 COVERS ONE BAUD INTERVAL OF TIME WHILE THE *
C* DIMENSION 62 REPRESENTS VOLTAGES RANGING FROM -1.2 TO +1.2   *
C* VOLTS.                                                        *
C*****************************************************************
      INTEGER STEP,BLANK,DASH,POINT,CH(32),ORD(13)
      DIMENSION MAT(130,61),ST(4097),X(4095)
      DATA (CH(I),I=1,32)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,
     +1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,
     +1HS,1HT,1HU,1HV,1HW/
      DATA BLANK/1H /,DASH/1H-/,POINT/1H./
      DATA (ORD(I),I=1,13)/2H12,2H10,1H8,1H6,1H4,1H2,1H0,1H2,1H4,1H6,
     +1H8,2H10,2H12/
      NST=NSB*MPS
      REWIND 2
      I=0
   10 I=I+1
      READ(2) ST(I)
      IF(I.GE.4097) GO TO 20
      IF(EOF(2)) 20,10
   20 JX=1
      STEP=128/NSB
      START=3*ST(1)-3*ST(2)+ST(3)
      ST(NST+1)=ST(NST-2)-3*ST(NST-1)+3*ST(NST)
      REWIND 3
      JG=NSB+1
      WRITE(3) START,(ST(J),J=1,JG)
      DO 11 N=2,MPS
         JA=(N-1)*NSB
         JB=JA+NSB+1
   11    WRITE(3) (ST(J),J=JA,JB)
      REWIND 3
      DO 22 J=1,130
         DO 22 K=1,61
   22       MAT(J,K)=BLANK
      DO 33 L=1,61,5
         K=L/5+1
         MAT(130,L)=ORD(K)
         DO 33 J=1,129
   33       MAT(J,L)=DASH
      DO 44 L=1,61
         MAT(1,L)=MAT(14,L)=MAT(27,L)=MAT(39,L)=MAT(52,L)=POINT
         MAT(65,L)=MAT(78,L)=MAT(91,L)=MAT(103,L)=MAT(115,L)=POINT
   44    MAT(129,L)=POINT
      JB=(NSB+1)*STEP+1
      INT=STEP-1
      DO 77 N=1,MPS
         READ(3) (X(J),J=1,JB,STEP)
         DO 55 J=1,NSB
            I1=(J-1)*STEP+1
            I2=I1+STEP
            I3=I2+STEP
            X1213=(I1-I2)*(I1-I3)
            X2123=(I2-I1)*(I2-I3)
            X3132=(I3-I1)*(I3-I2)
            DO 55 K=1,INT
               L=I1+K
               A=(L-I2)*(L-I3)*X(I1)/X1213
               B=(L-I1)*(L-I3)*X(I2)/X2123
               C=(L-I1)*(L-I2)*X(I3)/X3132
   55          X(L)=A+B+C
         NCH=MOD(N,32)
         IF(NCH.EQ.0) NCH=32
         DO 66 J=1,129,JX
            DO=(X(J)+1.2)*25.+6.
            LO=DO
            IF(DO-LO.GE.0.5) LO=LO+1
            IF(LO.LT.1) LO=1
            IF(LO.GT.61) LO=61
            LO=62-LO
   66       MAT(J,LO)=CH(NCH)
   77 CONTINUE
      PRINT 1
    1 FORMAT(1H1,1H0,11X,3H.1T,10X,3H.2T,9X,3H.3T,10X,3H-.4T,10X,
     +3H.5T,10X,3H.6T,10X,3H.7T,9X,3H.8T,10X,3H.9T,11X,1HT)
      PRINT 2,MAT
    2 FORMAT(1X,129A1,A2)
      RETURN
      END
```

```
      SUBROUTINE FILTER(SIGNAL,TF,N,M,LDIM)
C*************************************************************************
C* THIS SUBROUTINE PERFORMS THE OPERATION OF FOURIER TRANSFORMATION ON *
C* THE SIGNAL AND MULTIPLYING IT WITH THE FILTER TRANSFER FUNCTION.    *
C* THE RESULT IS RETURNED INTO THE ARRAY 'SIGNAL'.                     *
C*************************************************************************
      DIMENSION SIGNAL(LDIM),TF(LDIM),IWK(20)
      COMPLEX A(512)
      N1=M-1
C
C THE REAL ARRAY 'SIGNAL' IS TRANSFERRED INTO A COMPLEX ARRAY 'A'
C BY SUBROUTINE TRANSF. THE LIBRARY SUBROUTINE FFT2C
C PERFORMS THE FAST FOURIER TRANSFORMATION ON SIGNAL ARRAY 'A'.
C
      CALL TRANSF(A,SIGNAL,N,0,LDIM)
      CALL FFT2C(A,N1,IWK)
      DO 11 I=1,N
   11 A(I)=CONJG(A(I))
C *
C THE SIGNAL IS MULTIPLIED BY THE FILTER TRANSFER FUNCTION.
C
      DO 22 I=1,N
         II=I*2
         IR=II-1
         SIGNAL(IR)=TF(IR)*REAL(A(I))-TF(II)*AIMAG(A(I))
         SIGNAL(II)=TF(II)*REAL(A(I))+TF(IR)*AIMAG(A(I))
   22    A(I)=CMPLX(SIGNAL(IR),SIGNAL(II))
C
C INVERSE FFT IS PERFORMED ON COMPLEX ARRAY 'A'.
C
      CALL FFT2C(A,N1,IWK)
      DO 33 I=1,N
   33    A(I)=CONJG(A(I))/N
C
C THE COMPLEX ARRAY 'A' IS TRANSFERRED BACK INTO THE REAL ARRAY
C 'SIGNAL'.
C
      CALL TRANSF (A,SIGNAL,N,1,LDIM)
      RETURN
      END
```

```
      SUBROUTINE GDATA (N,M,X,XBAR,STD,D,SUMSQ)
C**********************************************************************
C* SUBROUTINE GDATA -                                                 *
C* -----------------                                                  *
C* PURPOSE                                                            *
C* GENERATE INDEPENDENT VARIABLES UP TO THE M-TH POWER (THE           *
C* HIGHEST DEGREE POLYNOMIAL SPECIFIED) AND COMPUTE MEANS,            *
C* STANDARD DEVIATIONS, AND CORRELATION COEFFICIENTS. THIS            *
C* SUBROUTINE IS NORMALLY CALLED BEFORE SUBROUTINES ORDER,            *
C* MINV AND MULTR IN THE PERFORMANCE OF A POLYNOMIAL                  *
C* REGRESSION.                                                        *
C*                                                                    *
C* USAGE                                                              *
C* CALL GDATA (N,M,X,XBAR,STD,D,SUMSQ)                                *
C*                                                                    *
C* DESCRIPTION OF PARAMETERS                                          *
C* N     - NUMBER OF OBSERVATIONS.                                    *
C* M     - THE HIGHEST DEGREE POLYNOMIAL TO BE FITTED.                *
C* X     - INPUT MATRIX (N BY M+1) . WHEN THE SUBROUTINE IS           *
C*         CALLED, DATA FOR THE INDEPENDENT VARIABLE ARE              *
C*         STORED IN THE FIRST COLUMN OF MATRIX X, AND DATA FOR       *
C*         THE DEPENDENT VARIABLE ARE STORED IN THE LAST              *
C*         COLUMN OF THE MATRIX. UPON RETURNING TO THE                *
C*         CALLING ROUTINE, GENERATED POWERS OF THE INDEPENDENT       *
C*         VARIABLE ARE STORED IN COLUMNS 2 THROUGH M.                *
C* XBAR  - OUTPUT VECTOR OF LENGTH M+1 CONTAINING MEANS OF            *
C*         INDEPENDENT AND DEPENDENT VARIABLES.                       *
C* STD   - OUTPUT VECTOR OF LENGTH M+1 CONTAINING STANDARD            *
C*         DEVIATIONS OF INDEPENDENT AND DEPENDENT VARIABLES.         *
C* D     - OUTPUT MATRIX (ONLY UPPER TRIANGULAR PORTION OF THE        *
C*         SYMMETRIC MATRIX OF M+1 BY M+1) CONTAINING CORRELA-        *
C*         TION COEFFICIENTS. (STORAGE MODE OF 1)                     *
C* SUMSQ - OUTPUT VECTOR OF LENGTH M+1 CONTAINING SUMS OF             *
C*         PRODUCTS OF DEVIATIONS FROM MEANS OF INDEPENDENT           *
C*         AND DEPENDENT VARIABLES.                                   *
C*                                                                    *
C* REMARKS                                                            *
C* N MUST BE GREATER THAN M+1.                                        *
C* IF M IS EQUAL TO 9 OR GREATER, SINGLE PRECISION MAY NOT BE         *
C* SUFFICIENT TO GIVE SATISFACTORY COMPUTATIONAL RESULTS.             *
C*                                                                    *
C* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                      *
C* NONE                                                               *
C*                                                                    *
C* METHOD                                                             *
C* REFER TO B. OSTLE, "STATISTICS IN RESEARCH", THE IOWA STATE        *
C* COLLEGE PRESS, 1954, CHAPTER 6.                                    *
C*                                                                    *
C**********************************************************************
```

```
          DIMENSION X(1),XBAR(1),STD(1),D(1),SUMSQ(1)
C
C...............................................................
C
C   IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C   C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C   STATEMENT WHICH FOLLOWS.
C
          DOUBLE PRECISION STD,D,X,XBAR,SUMSQ
C
C   THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C   APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C   ROUTINE.
C
C   THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C   CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT AND ABS IN
C   STATEMENT 180 MUST BE CHANGED TO DSQRT AND DABS.
C
C...............................................................
C
C   GENERATE INDEPENDENT VARIABLES
C
C 900 FORMAT (/,I3,//,(1X,10F10.4))
          L1=0
          DO 100 I=2,M
          L1=L1+N
          DO 100 J=1,N
          L=L1+J
          K=L-N
      100 X(L)=X(K)*X(J)
C
C   CALCULATE MEANS
C
          MM=M+1
          DF=N
          L=0
          DO 115 I=1,MM
          XBAR(I)=0.0
          DO 110 J=1,N
          L=L+1
      110 XBAR(I)=XBAR(I)+X(L)
      115 XBAR(I)=XBAR(I)/DF
C
C         JK=13
C         WRITE (6,900) JK,(X(JL),JL=1,40)
          DO 130 I=1,MM
      130 STD(I)=0.0
C
C   CALCULATE SUMS OF CROSS-PRODUCTS OF DEVIATIONS
C
          L=((MM+1)*MM)/2
          DO 150 I=1,L
      150 D(I)=0.0
          DO 170 K=1,N
          L=0
          DO 170 J=1,MM
          L2=N*(J-1)+K
          T2=X(L2)-XBAR(J)
          STD(J)=STD(J)+T2
          DO 170 I=1,J
          L1=N*(I-1)+K
          T1=X(L1)-XBAR(I)
          L=L+1
      170 D(L)=D(L)+T1*T2
C         JK=14
C         WRITE (6,900) JK,(X(JL),JL=1,40)
          L=0
          DO 175 J=1,MM
          DO 175 I=1,J
          L=L+1
      175 D(L)=D(L)-STD(I)*STD(J)/DF
          L=0
          DO 180 I=1,MM
          L=L+I
          SUMSQ(I)=D(L)
      180 STD(I)=DSQRT(DABS(D(L)))
C
C   CALCULATE CORRELATION COEFFICIENTS
C
C         JK=15
C         WRITE (6,900) JK,(X(JL),JL=1,40)
          L=0
          DO 190 J=1,MM
          DO 190 I=1,J
          L=L+1
      190 D(L)=D(L)/(STD(I)*STD(J))
```

```
C
C  CALCULATE STANDARD DEVIATIONS
C
C      JK=16
C      WRITE (6,900) JK,(X(JL),JL=1,40)
       DF=SQRT(DF-1.0)
       DO 200 I=1,MM
   200 STD(I)=STD(I)/DF
       RETURN
       END
```

```
       SUBROUTINE HISTOG
*************************************************************
*  THIS SUB-OUTINE SCANS A SET OF DATA POINTS STORED IN     *
*  TAPE2 AND PRINTS A HISTOGRAM OF THE DENSITY FUNCTION     *
*  OF THE DATA POINTS. THE INTERVAL, AB, BETWEEN ADJACENT   *
*  BARS OF THE HISTOGRAM MUST BE SPECIFIED AS INPUT DATA.   *
*************************************************************
       DIMENSION T(100),U(100),A(504),W(127),I+(50),X(100),Y(16384)
       DIMENSION IOPT(4),CRIT(4),STAT(5),PAR(2)
       REAL MEAN,MIN,MAX
       REWIND 2
       DO 11 I=1,100
    11    T(I)=U(I)=0.
       SQ2PI=SQRT(8.*ATAN2(1.,1.))
       MIN=MAX=SUM=SUMS=SUMSQ=0.
       K=1
    10 READ(2) Y(K)
       IF(EOF(2)) 20,15
    15 IF(MIN.GE.Y(K)) MIN=Y(K)
       IF(MAX.LE.Y(K)) MAX=Y(K)
       K=K+1
       IF(K.GT.16384) 20,10
    20 K=K-1
       DT=(MAX-MIN)/100.
       DO 33 I=I,K
          C=MIN
          DO 22 J=1,100
             L=J
             IF(Y(I).GE.C.AND.Y(I).LT.C+DT) GO TO 30
    22       C=C+DT
          GO TO 33
    30    T(L)=T(L)+1.
    33 CONTINUE
       DO 44 I=1,100
          SUM=SUM+T(I)
          SUMS=SUMS+T(I)*FLOAT(I)
    44    SUMSQ=SUMSQ+T(I)*FLOAT(I*I)
       MEAN=SUMS/SUM
       VAR=(SUMSQ-SUMS*SUMS/SUM)/(SUM-1)
       DEV=SQRT(VAR)
       PRINT *," MEAN = ",MEAN,",  VARIANCE = ",VAR,",  DEVIATION = ",
      +DEV
       DO 55 I=1,100
          X(I)=FLOAT(I)
          U(I)=SUM*(EXP(-(X(I)-MEAN)*(X(I)-MEAN)/(2.*VAR)))/(DEV*SQ2PI)
          N=U(I)
          IF(U(I)-FLOAT(N).GE.0.5) N=N+1
    55    U(I)=FLOAT(N)
       PRINT *," DT = ",DT
       PRINT *," ABSCISSA SCALE "
       PRINT 1,(X(I),I=1,100)
       PRINT *," SUM =",SUM,",  SUMS = ",SUMS,",  SUMSQ = ",SUMSQ
       PRINT *," FREQUENCY OF OCCURRENCE OF DATA VALUES ".
       PRINT 1,(T(I),I=1,100)
       PRINT *," ORDINATES OF GAUSSIAN PDF CURVE "
       PRINT 1,(U(I),I=1,100)
       CALL USHIST(T,U,100,0,A,W,1d,IER)
       PRINT 2,SUM,MEAN,VAR,DEV,MIN,MAX,DT
       READ *,(IOPT(I),I=1,4),(CRIT(I),I=1,4)
       PRINT *," IOPT(1 TO 4) = ",IOPT,",  CRIT(1 TO 4) = ",CRIT
       CALL GEMSON(Y,K,IOPT,CRIT,D,PAR,STAT,N,IER)
       PRINT *," PAR (1 TO 2) = ",PAR,",  # DEG. FREEDOM = ",N
       PRINT *," STAT (1 TO 5) = ",STAT
     1 FORMAT(10(2X,F11.2))
     2 FORMAT(1H0,20X,*IN THE ABOVE DISTRIBUTION,*//30X,
      +*NO. SAMPLE VALUES*,T70,* = *,F10.3/30X,*SAMPLE MEAN*,
      +T70,* = *,F10.3/30X,*SAMPLE VARIANCE*,T70,* = *,
      +F10.3/30X,*STANDARD DEVIATION*,T70,* = *,F10.3/30X,
      +*LIMIT OF NEGATIVE SIDE OF ABSCISSA*,T70,* = *,F10.3/30X,
      +*LIMIT OF POSITIVE SIDE OF ABSCISSA*,T70,* = *,F10.3/30X,
      +*ELEM. DIST. BETWEEN HISTOGRAM BARS*,T70,* = *,F10.3//)
       STOP
       END
```

```
      SUBROUTINE IDFIL(SBANDW,FBANDW,TF,LDIM,L)
*********************************************************
*  THIS SUBROUTINE GENERATES THE TRANSFER FUNCTION      *
*  OF AN IDEAL LOW PASS FILTER.                         *
*********************************************************
      DIMENSION TF(LDIM)
      NH=LDIM/2
      ND=LDIM/4
      PRINT 1
      DW=SBANDW/FLOAT(NH)/FBANDW*2.
      IF(L.EQ.1) DW=DW*2.
      W=0.
      DO 11 I=1,ND
         X=0.
         IF(W.LE.1.) X=1.
         I2=I*2
         I1=I2-1
         TF(I1)=X
         TF(I2)=0.
         I2=2*(NH-I+2)
         I1=I2-1
         TF(I1)=X
         TF(I2)=0.
   11    W=W+DW
    1 FORMAT(1H1,4X,*AN IDEAL FILTER IS USED.*//)
      RETURN
      END
```

```
      SUBROUTINE LOAD(DATA,KKK,LSAMPL,NX,I,JLAST,JTAP,LDIM)
************************************************************
*  THIS SUBROUTINE GENERATES THE PSEUDO-RANDOM-BINARY-     *
*  SEQUENCES AND LOADS THEM INTO THE ARRAY DATA.           *
*        KKK = NO. SIGNALS BEING LOADED.                   *
*     LSAMPL = NO. SAMPLES PER SYMBOL.                     *
*         NX = NAME OF THE SHIFT REGISTER ARRAY THAT       *
*              GENERATES PN-SEQUENCES.                     *
*          I = INDEX INDICATING THE SHIFT REGISTER OUTPUT  *
*              POSITIONS AFTER SHIFTING.                   *
*      JLAST = LENGTH OF SHIFT REGISTER.                   *
*       JTAP = TAP LOCATION.                               *
*       DATA = NAME OF ARRAY INTO WHICH THE SIGNAL IS      *
*              LOADED.                                     *
*       LDIM = DIMENSION OF ARRAY DATA.                    *
*         J4 = STARTING POSITION FOR FILLING *DATA* WITH   *
*              ZEROS.                                      *
************************************************************
      DIMENSION DATA(LDIM),NX(JLAST)
      JA=2*LSAMPL
      DO 22 K=1,KKK
C
C     GENERATING ONE SYMBOL.
C
         IF(I.EQ.JLAST) I=0
         I=I+1
         J=I-JTAP
         IF(I.LE.JTAP) J=J+JLAST
         IX=NX(J)*NX(I)
         NX(I)=IX
C
C     LOAD SIGNAL INTO ARRAY DATA.
C
         JB=K*JA
         J1=JB-JA+1
         J2=JB-1
         DO 11 J3=J1,J2,2
            DATA(J3)=IX
   11       DATA(J3+1)=0.
   22 CONTINUE
C
C     FILL THE REST OF ARRAY DATA WITH ZEROS FOR SPILLOVER EFFECT
C     OF THE FILTER.
C
      J4=J2+2
      DO 33 J3=J4,LDIM
   33    DATA(J3)=0.
      RETURN
      END
```

```
      SUBROUTINE L1(M,N,M2,N2,A,B,TOLER,X,E,S)
C*********************************************************************
C*                                                                   *
C*  THIS SUBROUTINE USES A MODIFICATION OF THE SIMPLEX METHOD        *
C*  OF LINEAR PROGRAMMING TO CALCULATE AN L1 SOLUTION TO AN          *
C*  OVER-DETERMINED SYSTEM OF LINEAR EQUATIONS.                      *
C*                                                                   *
C*  DESCRIPTION OF PARAMETERS                                        *
C*  -------------------------                                        *
C*                                                                   *
C*       M = NUMBER OF EQUATIONS.                                    *
C*       N = NUMBER OF UNKNOWNS (M.GE.N).                            *
C*      M2 = SET EQUAL TO M+2 FOR ADJUSTABLE DIMENSIONS.             *
C*      N2 = SET EQUAL TO N+2 FOR ADJUSTABLE DIMENSIONS.             *
C*       A = TWO DIMENSIONAL REAL ARRAY OF SIZE (M2,N2).             *
C*           ON ENTRY, THE COEFFICIENTS OF THE MATRIX MUST BE        *
C*           STORED IN THE FIRST M ROWS AND N COLUMNS OF A.          *
C*           THESE VALUES ARE DESTROYED BY THE SUBROUTINE.           *
C*       B = ONE DIMENSIONAL REAL ARRAY OF SIZE M.                   *
C*           ON ENTRY, B MUST CONTAIN THE RIGHT HAND SIDE OF         *
C*           THE EQUATIONS. THESE VALUES ARE DESTROYED BY THE        *
C*           SUBROUTINE.                                             *
C*  TOLER = A SMALL POSITIVE TOLERANCE.                              *
C*           EMPIRICAL EVIDENCE SUGGESTS TOLER=10**(-D*2/3)          *
C*           WHERE D REPRESENTS THE NUMBER OF DECIMAL DIGITS         *
C*           OF ACCURACY AVAILABLE (SEE DESCRIPTION).                *
C*       X = ONE DIMENSIONAL REAL ARRAY OF SIZE N.                   *
C*           ON EXIT, THIS ARRAY CONTAINS A SOLUTION TO THE          *
C*           L1 PROBLEM.                                             *
C*       E = ONE DIMENSIONAL REAL ARRAY OF SIZE M.                   *
C*           ON EXIT, THIS ARRAY CONTAINS THE RESIDUALS IN           *
C*           THE EQUATIONS.                                          *
C*       S = INTEGER ARRAY OF SIZE M USED FOR WORKSPACE.             *
C*                                                                   *
C*  ON EXIT FROM THE SUBROUTINE, THE ARRAY "A" CONTAINS THE          *
C*  FOLLOWING INFORMATION.                                           *
C*                                                                   *
C*  A(M+1,N+1)  =  THE MINIMUM SUM OF THE ABSOLUTE VALUES OF         *
C*                 THE RESIDUALS.                                    *
C*  A(M+1,N+2)  =  THE RANK OF THE MATRIX OF COEFFICIENTS.           *
C*  A(M+2,N+1)  =  EXIT CODES WITH VALUES,                           *
C*                     0 - OPTIMAL SOLUTION WHICH IS PROBABLY        *
C*                         NON-UNIQUE (SEE DESCRIPTION).             *
C*                     1 - UNIQUE OPTIMAL SOLUTION.                  *
C*                     2 - CALCULATIONS TERMINATED PREMATURELY       *
C*                         DUE TO ROUNDING ERRORS.                   *
C*  A(M+2,N+2)  =  NUMBER OF SIMPLEX ITERATIONS PERFORMED.           *
C*                                                                   *
C*  REFERENCE: BARRODALE, I. AND F.D.K. ROBERTS, "SOLUTION OF AN     *
C*  OVERDETERMINED SYSTEM OF EQUATIONS IN THE L1 NORM[F4]," COMMU-   *
C*  NICATIONS OF THE ACM, VOL.17, NO.6, PP.319-320, JUNE 1974.       *
C*********************************************************************
      DOUBLE PRECISION SUM
      REAL MIN,MAX,A(M2,N2),X(N),E(M),B(M)
      INTEGER OUT,S(M)
      LOGICAL STAGE,TEST
C
C  BIG MUST BE SET EQUAL TO ANY VERY LARGE REAL CONSTANT.
C  ITS VALUE HERE IS APPROPRIATE FOR THE IBM 370.
C
      DATA BIG/1.E75/
C
C  INITIALIZATION.
C
      M1=M+1
      N1=N+1
      DO 10 J=1,N
        A(M2,J)=J
   10   X(J)=0.
      DO 40 I=1,M
        A(I,N2)=N+I
        A(I,N1)=B(I)
        IF(B(I).GE.0.) GO TO 30
        DO 20 J=1,N2
   20     A(I,J)=-A(I,J)
   30   E(I)=0.
   40 CONTINUE
C
C  COMPUTE MARGINAL COSTS.
C
      DO 60 J=1,N1
        SUM=0.D0
        DO 50 I=1,M
   50     SUM=SUM+A(I,J)
   60   A(M1,J)=SUM
```

```
C
C  STAGE I.
C  DETERMINE THE VECTOR TO ENTER THE BASIS.
C
      STAGE=.TRUE.
      KOUNT=0
      KR=1
      KL=1
   70 MAX=-1.
      DO 80 J=KR,N
        IF(ABS(A(M2,J)).GT.N) GO TO 80
        D=ABS(A(M1,J))
        IF(D.LE.MAX) GO TO 80
        MAX=D
        IN=J
   80 CONTINUE
      IF(A(M1,IN).GE.0) GO TO 100
      DO 90 I=1,M2
   90   A(I,IN)=-A(I,IN)
C
C  DETERMINE THE VECTOR TO LEAVE THE BASIS.
C
  100 K=0
      DO 110 I=KL,M
        D=A(I,IN)
        IF(D.LE.TOLER) GO TO 110
        K=K+1
        B(K)=A(I,N1)/D
        S(K)=I
        TEST=.TRUE.
  110 CONTINUE
  120 IF(K.GT.0) GO TO 130
      TEST=.FALSE.
      GO TO 150
  130 MIN=BIG
      DO 140 I=1,K
        IF(B(I).GE.MIN) GO TO 140
        J=I
        MIN=B(I)
        OUT=S(I)
  140 CONTINUE
      B(J)=B(K)
      S(J)=S(K)
      K=K-1
C
C  CHECK FOR LINEAR DEPENDENCE IN STAGE I.
C
  150 IF(TEST.OR..NOT.STAGE) GO TO 170
      DO 160 I=1,M2
        D=A(I,KR)
        A(I,KR)=A(I,IN)
  160   A(I,IN)=D
      KR=KR+1
      GO TO 260
  170 IF(TEST) GO TO 180
      A(M2,N1)=2.
      GO TO 350
  180 PIVOT=A(OUT,IN)
      IF(A(M1,IN)-PIVOT-PIVOT.LE.TOLER) GO TO 200
      DO 190 J=KR,N1
        D=A(OUT,J)
        A(M1,J)=A(M1,J)-D-D
  190   A(OUT,J)=-D
      A(OUT,N2)=-A(OUT,N2)
      GO TO 120
C
C  PIVOT ON A(OUT,IN).
C
  200 DO 210 J=KR,N1
        IF(J.EQ.IN) GO TO 210
        A(OUT,J)=A(OUT,J)/PIVOT
  210 CONTINUE
      DO 230 I=1,M1
        IF(I.EQ.OUT) GO TO 230
        D=A(I,IN)
        DO 220 J=KR,N1
          IF(J.EQ.IN) GO TO 220
          A(I,J)=A(I,J)-D*A(OUT,J)
  220   CONTINUE
  230 CONTINUE
      DO 240 I=1,M1
        IF(I.EQ.OUT) GO TO 240
        A(I,IN)=-A(I,IN)/PIVOT
  240 CONTINUE
      A(OUT,IN)=1./PIVOT
      D=A(OUT,N2)
```

```
      A(OUT,N2)=A(M2,IN)
      A(M2,IN)=D
      KOUNT=KOUNT+1
      IF(.NOT.STAGE) GO TO 270
C
C  INTERCHANGE ROWS IN STAGE I.
C
      KL=KL+1
      DO 250 J=KR,N2
        D=A(OUT,J)
        A(OUT,J)=A(KOUNT,J)
  250   A(KOUNT,J)=D
  260 IF(KOUNT+KR.NE.M1) GO TO 70
C
C  STAGE II.
C
      STAGE=.FALSE.
C
C  DETERMINE THE VECTOR TO ENTER THE BASIS.
C
  270 MAX=-BIG
      DO 290 J=KR,N
        D=A(M1,J)
        IF(D.GE.0.) GO TO 280
        IF(D.GT.(-2.)) GO TO 290
        D=-D-2.
  280   IF(D.LE.MAX) GO TO 290
        MAX=D
        IN=J
  290 CONTINUE
      IF(MAX.LE.TOLER) GO TO 310
      IF(A(M1,IN).GT.0.) GO TO 100
      DO 300 I=1,M2
  300   A(I,IN)=-A(I,IN)
      A(M1,IN)=A(M1,IN)-2.
      GO TO 100
C
C  PREPARE OUTPUT.
C
  310 L=KL-1
      DO 330 I=1,L
        IF(A(I,N1).GE.0.) GO TO 330
        DO 320 J=KR,N2
  320     A(I,J)=-A(I,J)
  330 CONTINUE
      A(M2,N1)=0.
      IF(KR.NE.1) GO TO 350
      DO 340 J=1,N
        D=ABS(A(M1,J))
  340   IF(D.LE.TOLER .OR. 2.-D.LE.TOLER) GO TO 350
      A(M2,N1)=1.
  350 DO 380 I=1,M
        K=A(I,N2)
        D=A(I,N1)
        IF(K.GT.0) GO TO 360
        K=-K
        D=-D
  360   IF(I.GE.KL) GO TO 370
        X(K)=D
        GO TO 380
  370   K=K-N
        E(K)=D
  380 CONTINUE
      A(M2,N2)=KOUNT
      A(M1,N2)=N1-KR
      SUM=0.D0
      DO 390 I=KL,M
  390   SUM=SUM+A(I,N1)
      RETURN
      END
```

```
      SUBROUTINE MEASFIL(SBANDW,FBANDW,TF,LDIM)
***************************************************************
*  MEASURED AMPLITUDE AND PHASE CHARACTERISTICS OF A LOW PASS   *
*  FILTER ARE TRANSFORMED INTO A COMPLEX TRANSFER FUNCTION. ALL *
*  THE COEFFICIENTS AND THE CONSTANT OF 13-DEGREE POLYNOMIALS   *
*  OF THESE CHARACTERISTICS MUST BE GIVEN AS INPUT DATA. THEY   *
*  ARE STORED IN ARRAYS "CA" AND "CP". CA(14) AND CP(14) CONTA- *
*  IN THE TWO CONSTANTS OF THESE POLYNOMIALS.                   *
***************************************************************
      DIMENSION TF(LDIM),CA(14),CP(14),A(4)
      READ 1,(A(I),I=1,4)
      READ *,(CA(I),I=1,14)
      READ *,(CP(I),I=1,14)
      NH=LDIM/2
      NO=LDIM/4
      DW=SBANDW/FLOAT(NH)
      PRINT 2,(A(I),I=1,4)
      PRINT 3,CA(14),(CA(I),I=1,13)
      PRINT 4,CP(14),(CP(I),I=1,13)
C
C  BLANK THE TRANSFER FUNCTION ARRAY TF.
C
      DO 11 I=1,LDIM
   11    TF(I)=0.
      DO 22 I=2,NO
         W=DW*FLOAT(I-1)
         F1=W
         F2=F1*F1
         F3=F2*F1
         F4=F2*F2
         F5=F4*F1
         F6=F3*F3
         F7=F6*F1
         F8=F4*F4
         F9=F8*F1
         F10=F5*F5
         F11=F10*F1
         F12=F6*F6
         F13=F12*F1
         X=CA(14)+CA(1)*F1+CA(2)*F2+CA(3)*F3+CA(4)*F4+CA(5)*F5
     *           +CA(6)*F6+CA(7)*F7+CA(8)*F8+CA(9)*F9+CA(10)*F10
     *           +CA(11)*F11+CA(12)*F12+CA(13)*F13
         Y=       CP(1)*F1+CP(2)*F2+CP(3)*F3+CP(4)*F4+CP(5)*F5
     *           +CP(6)*F6+CP(7)*F7+CP(8)*F8+CP(9)*F9+CP(10)*F10
     *           +CP(11)*F11+CP(12)*F12+CP(13)*F13
         IF(X.LT.0.) GO TO 10
         Y=Y/57.2957795
         XX=X*COS(Y)
         YY=X*SIN(Y)
         XX=XX/CA(14)
         YY=YY/CA(14)
         I2=I*2
         I1=I2-1
         TF(I1)=XX
         TF(I2)=YY
         I2=2*(NH-I+2)
         I1=I2-1
         TF(I1)=XX
   22    TF(I2)=-YY
   10 TF(1)=1.
      TF(2)=TF(NH+1)=TF(NH+2)=0.
    1 FORMAT(4A10)
    2 FORMAT(1H1,4X,'A PRACTICAL FILTER IS USED. THE FILTER TYPE IS,'/
     +5X,4A10//5X,'ITS POLYNOMIAL COEFFICIENTS ARE,'/)
    3 FORMAT(//' BASEBAND TRANSMIT AMPLITUDE (LINEAR) -',
     +/5X,' DESCRIPTION -'
     +/5X,' COEFFICIENTS-',' A( 0) ',1PD12.4,5X,'A( 1) ',D12.4,
     +5X,'A( 2) ',D12.4/19X,'A( 3)',D13.4,5X,'A( 4) ',D12.4,
     +5X,'A( 5) ',D12.4/19X,'A( 6)',D13.4,5X,'A( 7) ',D12.4,
     +5X,'A( 8) ',D12.4/19X,'A( 9)',D13.4,5X,'A(10) ',D12.4,
     +5X,'A(11) ',D12.4/19X,'A(12)',D13.4,5X,'A(13) ',D12.4)
    4 FORMAT(//' BASEBAND TRANSMIT PHASE (DEG)-',
     +/5X,' DESCRIPTION -'
     +/5X,' COEFFICIENTS-','P( 0) ',1PD12.4,5X,'P( 1)',D13.4,
     +5X,'P( 2) ',D12.4/19X,'P( 3)',D13.4,5X,'P( 4) ',D12.4,
     +5X,'P( 5) ',D12.4/19X,'P( 6)',D13.4,5X,'P( 7) ',D12.4,
     +5X,'P( 8) ',D12.4/19X,'P( 9)',D13.4,5X,'P(10) ',D12.4,
     +5X,'P(11) ',D12.4/19X,'P(12)',D13.4,5X,'P(13) ',D12.4)
      RETURN
      END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
*****************************************************************
*                                                               *
*                                                               *
*     SUBROUTINE MINV                                           *
*                                                               *
*     PURPOSE                                                   *
*     INVERT A MATRIX                                           *
*                                                               *
*     USAGE                                                     *
*     CALL MINV(A,N,D,L,M)                                      *
*                                                               *
*     DESCRIPTION OF PARAMETERS                                 *
*     A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY *
*         RESULTANT INVERSE.                                    *
*     N - ORDER OF MATRIX A                                     *
*     D - RESULTANT DETERMINANT                                 *
*     L - WORK VECTOR OF LENGTH N                               *
*     M - WORK VECTOR OF LENGTH N                               *
*                                                               *
*     REMARKS                                                   *
*     MATRIX A MUST BE A GENERAL MATRIX                         *
*                                                               *
*     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED             *
*     NONE                                                      *
*                                                               *
*     METHOD                                                    *
*     THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT *
*     IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT  *
*     THE MATRIX IS SINGULAR.                                   *
*                                                               *
*****************************************************************
      DIMENSION A(1),L(1),M(1)
C
C................................................................
C
C     IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C     C IN COLUMN    1   SHOULD  BE REMOVED FROM THE DOUBLE PRECISION
C     STATEMENT WHICH FOLLOWS.
C
      DOUBLE PRECISION A,D,BIGA,HOLD,L,M
C
C     THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C     APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C     ROUTINE.
C
C     THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C     CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C     10 MUST BE  CHANGED TO DABS.
C
C................................................................
C
C     SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ)))15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C     INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
```

```
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=A(IK)*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
      SUBROUTINE MULTR (N,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS)
C.....................................................................
C
C     SUBROUTINE MULTR
C
C     PURPOSE
C        PERFORM A MULTIPLE LINEAR REGRESSION ANALYSIS FOR A DEPEN-
C        DENT VARIABLE AND A SET OF INDEPENDENT VARIABLES.  THIS
C        SUBROUTINE IS NORMALLY USED IN THE PERFORMANCE OF MULTIPLE
C        AND POLYNOMIAL REGRESSION ANALYSES.
C
C     USAGE
C        CALL MULTR (N,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS)
C
C     DESCRIPTION OF PARAMETERS
C        N     - NUMBER OF OBSERVATIONS.
C        K     - NUMBER OF INDEPENDENT VARIABLES IN THIS REGRESSION.
C        XBAR  - INPUT VECTOR OF LENGTH M CONTAINING MEANS OF ALL
C                VARIABLES. M IS NUMBER OF VARIABLES IN OBSERVATIONS.
C        STD   - INPUT VECTOR OF LENGTH M CONTAINING STANDARD DEVI-
C                ATIONS OF ALL VARIABLES.
C        D     - INPUT VECTOR OF LENGTH M CONTAINING THE DIAGONAL OF
C                THE MATRIX OF SUMS OF CROSS-PRODUCTS OF DEVIATIONS
C                FROM MEANS FOR ALL VARIABLES.
C        RX    - INPUT MATRIX (K X K) CONTAINING THE INVERSE OF
C                INTERCORRELATIONS AMONG INDEPENDENT VARIABLES.
C        RY    - INPUT VECTOR OF LENGTH K CONTAINING INTERCORRELA-
C                TIONS OF INDEPENDENT VARIABLES WITH DEPENDENT
C                VARIABLE.
C        ISAVE - INPUT VECTOR OF LENGTH K+1 CONTAINING SUBSCRIPTS OF
C                INDEPENDENT VARIABLES IN ASCENDING ORDER.  THE
C                SUBSCRIPT OF THE DEPENDENT VARIABLE IS STORED IN
C                THE LAST, K+1, POSITION.
C        B     - OUTPUT VECTOR OF LENGTH K CONTAINING REGRESSION
C                COEFFICIENTS.
C        SB    - OUTPUT VECTOR OF LENGTH K CONTAINING STANDARD
C                DEVIATIONS OF REGRESSION COEFFICIENTS.
C        T     - OUTPUT VECTOR OF LENGTH K CONTAINING T-VALUES.
C        ANS   - OUTPUT VECTOR OF LENGTH 10 CONTAINING THE FOLLOWING
C                INFORMATION.
C                ANS(1)  INTERCEPT
C                ANS(2)  MULTIPLE CORRELATION COEFFICIENT
C                ANS(3)  STANDARD ERROR OF ESTIMATE
C                ANS(4)  SUM OF SQUARES ATTRIBUTABLE TO REGRES-
C                        SION (SSAR)
C                ANS(5)  DEGREES OF FREEDOM ASSOCIATED WITH SSAR
C                ANS(6)  MEAN SQUARE OF SSAR
C                ANS(7)  SUM OF SQUARES OF DEVIATIONS FROM REGRES-
C                        SION (SSDR)
C                ANS(8)  DEGREES OF FREEDOM ASSOCIATED WITH SSDR
C                ANS(9)  MEAN SQUARE OF SSDR
C                ANS(10) F-VALUE
C
C     REMARKS
C        N MUST BE GREATER THAN K+1.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        THE GAUSS-JORDAN METHOD IS USED IN THE SOLUTION OF THE
C        NORMAL EQUATIONS.  REFER TO W. W. COOLEY AND P. R. LOHNES,
C        "MULTIVARIATE PROCEDURES FOR THE BEHAVIORAL SCIENCES",
C        JOHN WILEY AND SONS, 1962, CHAPTER 3, AND B. OSTLE,
C        "STATISTICS IN RESEARCH", THE IOWA STATE COLLEGE PRESS,
C        1954, CHAPTER 8.
C
C.....................................................................
C
      DIMENSION XBAR(1),STD(1),D(1),RX(1),RY(1),ISAVE(1),B(1),SB(1),
     1T(1),ANS(10)
C
C.....................................................................
C
C     IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C     C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C     STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION XBAR,STD,D,RX,RY,B,SB,T,ANS,RM,BO,SSAR,SSDR,
C    1SY,FN,FK,SSARM,SSDRM,F,ISAVE
C
C     THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C     APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C     ROUTINE.
C
```

```
C  THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C  CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT AND ABS IN
C  STATEMENTS 122, 125, AND 135 MUST BE CHANGED TO DSQRT AND DABS.
C
C.....................................................................
C
      MM=K+1
C
C  BETA WEIGHTS
C
      DO 100 J=1,K
  100 B(J)=0.0
      DO 110 J=1,K
      L1=K*(J-1)
      DO 110 I=1,K
      L=L1+I
  110 B(J)=B(J)+RY(I)*RX(L)
      RM=0.0
      BO=0.0
      L1=ISAVE(MM)
C
C  COEFFICIENT OF DETERMINATION
C
      DO 120 I=1,K
      RM=RM+B(I)*RY(I)
C
C  REGRESSION COEFFICIENTS
C
      L=ISAVE(I)
      B(I)=B(I)*(STD(L1)/STD(L))
C
C  INTERCEPT
C
  120 BO=BO+B(I)*XBAR(L)
      BO=XBAR(L1)-BO
C
C  SUM OF SQUARES ATTRIBUTABLE TO REGRESSION
C
      SSAR=RM*D(L1)
C
C  MULTIPLE CORRELATION COEFFICIENT
C
  122 RM=DSQRT(DABS(RM))
C
C  SUM OF SQUARES OF DEVIATIONS FROM REGRESSION
C
      SSDR=D(L1)-SSAR
C
C  VARIANCE OF ESTIMATE
C
      FN=N-K-1
      SY=SSDR/FN
C
C  STANDARD DEVIATIONS OF REGRESSION COEFFICIENTS
C
      DO 130 J=1,K
      L1=K*(J-1)+J
      L=ISAVE(J)
  125 SB(J)=DSQRT(DABS((RX(L1)/D(L))*SY))
C
C  COMPUTED T-VALUES
C
  130 T(J)=B(J)/SB(J)
C
C  STANDARD ERROR OF ESTIMATE
C
  135 SY=DSQRT(DABS(SY))
C
C  F VALUE
C
      FK=K
      SSARM=SSAR/FK
      SSDRM=SSDR/FN
      F=SSARM/SSDRM
      ANS(1)=BO
      ANS(2)=RM
      ANS(3)=SY
      ANS(4)=SSAR
      ANS(5)=FK
      ANS(6)=SSARM
      ANS(7)=SSDR
      ANS(8)=FN
      ANS(9)=SSDRM
      ANS(10)=F
      RETURN
      END
```

```
      SUBROUTINE ORDER (M,R,NDEP,K,ISAVE,RX,RY)
*******************************************************************
*                                                                 *
*  SUBROUTINE ORDER                                               *
*                                                                 *
*  PURPOSE                                                        *
*  CONSTRUCT FROM A LARGER MATRIX OF CORRELATION COEFFICIENTS     *
*  A SUBSET MATRIX OF INTERCORRELATIONS AMONG INDEPENDENT         *
*  VARIABLES AND A VECTOR OF INTERCORRELATIONS OF INDEPENDENT     *
*  VARIABLES WITH DEPENDENT VARIABLE.  THIS SUBROUTINE IS         *
*  NORMALLY USED IN THE PERFORMANCE OF MULTIPLE AND POLYNOMIAL    *
*  REGRESSION ANALYSES.                                           *
*                                                                 *
*  USAGE                                                          *
*  CALL ORDER (M,R,NDEP,K,ISAVE,RX,RY)                            *
*                                                                 *
*  DESCRIPTION OF PARAMETERS                                      *
*  M      - NUMBER OF VARIABLES AND ORDER OF MATRIX R.            *
*  R      - INPUT MATRIX CONTAINING CORRELATION COEFFICIENTS.     *
*           THIS SUBROUTINE EXPECTS ONLY UPPER TRIANGULAR         *
*           PORTION OF THE SYMMETRIC MATRIX TO BE STORED (BY      *
*           COLUMN) IN R.  (STORAGE MODE OF 1)                    *
*  NDEP   - THE SUBSCRIPT NUMBER OF THE DEPENDENT VARIABLE.       *
*  K      - NUMBER OF INDEPENDENT VARIABLES TO BE INCLUDED        *
*           IN THE FORTHCOMING REGRESSION.                        *
*  ISAVE  - INPUT VECTOR OF LENGTH K+1 CONTAINING, IN ASCENDING   *
*           ORDER, THE SUBSCRIPT NUMBERS OF K INDEPENDENT         *
*           VARIABLES TO BE INCLUDED IN THE FORTHCOMING REGRES-   *
*           SION.                                                 *
*           UPON RETURNING TO THE CALLING ROUTINE, THIS VECTOR    *
*           CONTAINS, IN ADDITION, THE SUBSCRIPT NUMBER OF        *
*           THE DEPENDENT VARIABLE IN K+1 POSITION.               *
*  RX     - OUTPUT MATRIX (K X K) CONTAINING INTERCORRELATIONS    *
*           AMONG INDEPENDENT VARIABLES TO BE USED IN FORTH-      *
*           COMING REGRESSION.                                    *
*  RY     - OUTPUT VECTOR OF LENGTH K CONTAINING INTERCORRELA-    *
*           TIONS OF INDEPENDENT VARIABLES WITH DEPENDENT          *
*           VARIABLES.                                            *
*                                                                 *
*  REMARKS                                                        *
*  NONE                                                           *
*                                                                 *
*  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  *
*  NONE                                                           *
*                                                                 *
*  METHOD                                                         *
*  FROM THE SUBSCRIPT NUMBERS OF THE VARIABLES TO BE INCLUDED     *
*  IN THE FORTHCOMING REGRESSION, THE SUBROUTINE CONSTRUCTS THE   *
*  MATRIX RX AND THE VECTOR RY.                                   *
*                                                                 *
*******************************************************************
      DIMENSION R(1),ISAVE(1),RX(1),RY(1)
C
C.................................................................
C
C  IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C  C IN COLUMN  1  SHOULD  BE  REMOVED FROM THE DOUBLE PRECISION
C  STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION R,ISAVE,RX,RY
C
C  THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C  APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C  ROUTINE.
C
C.................................................................
C
C  COPY INTERCORRELATIONS OF INDEPENDENT VARIABLES
C  WITH DEPENDENT VARIABLE
C
      MM=0
      DO 130 J=1,K
      L2=ISAVE(J)
      IF(NDEP-L2) 122, 123, 123
  122 L=NDEP+(L2*L2-L2)/2
      GO TO 125
  123 L=L2+(NDEP*NDEP-NDEP)/2
  125 RY(J)=R(L)
```

```
      C
      C  COPY A SUBSET MATRIX OF INTERCORRELATIONS AMONG
      C  INDEPENDENT VARIABLES
      C
            DO 130 I=1,K
            L1=ISAVE(I)
            IF(L1-L2) 127, 128, 128
        127 L=L1+(L2*L2-L2)/2
            GO TO 129
        128 L=L2+(L1*L1-L1)/2
        129 MM=MM+1
        130 RX(MM)=R(L)
      C
      C  PLACE THE SUBSCRIPT NUMBER OF THE DEPENDENT
      C  VARIABLE IN ISAVE(K+1)
      C
            ISAVE(K+1)=NDEP
            RETURN
            END
```

```
            SUBROUTINE PLOT(NO,A,N,M,NL,NS)
      ................................................................
      .                                                              .
      .  SUBROUTINE PLOT                                             .
      .                                                              .
      .  PURPOSE                                                     .
      .  PLOT SEVERAL CROSS-VARIABLES VERSUS A BASE VARIABLE         .
      .                                                              .
      .  USAGE                                                       .
      .  CALL PLOT (NO,A,N,M,NL,NS)                                  .
      .                                                              .
      .  DESCRIPTION OF PARAMETERS                                   .
      .  NO - CHART NUMBER (3 DIGITS MAXIMUM)                        .
      .  A  - MATRIX OF DATA TO BE PLOTTED. FIRST COLUMN REPRESENTS  .
      .       BASE VARIABLE AND SUCCESSIVE COLUMNS ARE THE CROSS-    .
      .       VARIABLES (MAXIMUM IS 9).                              .
      .  N  - NUMBER OF ROWS IN MATRIX A                             .
      .  M  - NUMBER OF COLUMNS IN MATRIX A (EQUAL TO THE TOTAL      .
      .       NUMBER OF VARIABLES). MAXIMUM IS 10.                   .
      .  NL - NUMBER OF LINES IN THE PLOT. IF 0 IS SPECIFIED, 50     .
      .       LINES ARE USED.                                        .
      .  NS - CODE FOR SORTING THE BASE VARIABLE DATA IN ASCENDING   .
      .       ORDER                                                  .
      .         0  SORTING IS NOT NECESSARY (ALREADY IN ASCENDING    .
      .            ORDER).                                           .
      .         1  SORTING IS NECESSARY.                             .
      .                                                              .
      .  REMARKS                                                     .
      .  NONE                                                        .
      .                                                              .
      .  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED               .
      .  NONE                                                        .
      .                                                              .
      ................................................................
            DIMENSION OUT(101),YPR(11),ANG(9),A(1)
          1 FORMAT(1H1,T20,7H CHART ,I3,/T20,10H ----------//)
          2 FORMAT(1H ,F9.4,5X,101A1)
          3 FORMAT(1H )
          4 FORMAT(10H 123456789)
          5 FORMAT(10A1)
          7 FORMAT(1H0,12X,101H.---------+---------.---------.---------.------
           +---.---------.---------.---------.---------.---------.)
          8 FORMAT(1H0,5X,11F10.4)
            NLL=NL
            IF(NS) 16, 16, 10
      C
      C  SORT BASE VARIABLE DATA IN ASCENDING ORDER
      C
         10 DO 15 I=1,N
            DO 14 J=I,N
            IF(A(I)-A(J)) 14, 14, 11
         11 L=I-N
            LL=J-N
            DO 12 K=1,M
            L=L+N
            LL=LL+N
            F=A(L)
```

```
      A(L)=A(LL)
   12 A(LL)=F
   14 CONTINUE
   15 CONTINUE
C
C  TEST NLL
C
   16 IF(NLL) 20, 18, 20
   18 NLL=50
C
C  PRINT TITLE
C
   20 WRITE(6,1)NO
C
C  DEVELOP BLANK AND DIGITS FOR PRINTING
C
      DATA BLANK,ANG/1H ,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
C
C  FIND SCALE FOR BASE VARIABLE
C
      XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
C
C  FIND SCALE FOR CROSS-VARIABLES
C
      YMIN=1.0E38
       YMAX=-1.0E38
      M1=N+1
      M2=M*N
      DO 40 J=M1,M2
      IF(A(J)-YMIN) 28,26,26
   26 IF(A(J)-YMAX) 40,40,30
   28 YMIN=A(J)
      GO TO 26
   30 YMAX=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/100.0
C
C  FIND BASE VARIABLE PRINT POSITION
C
      XB=A(1)
      L=1
      MY=M-1
      DO 80 I=1,NLL
      F=I-1
      XPR=XB+F*XSCAL
      IF(A(L)-XPR) 50,50,70
C
C  FIND CROSS-VARIABLES
C
   50 DO 55 IX=1,101
   55 OUT(IX)=BLANK
      DO 60 J=1,MY
      LL=L+J*N
      JP=((A(LL)-YMIN)/YSCAL)+1.0
      OUT(JP)=ANG(J)
   60 CONTINUE
C
C  PRINT LINE AND CLEAR, OR SKIP
C
      WRITE(6,2)XPR,(OUT(IZ),IZ=1,101)
      L=L+1
      GO TO 80
   70 WRITE(6,3)
   80 CONTINUE
C
C  PRINT CROSS-VARIABLES NUMBERS
C
      WRITE(6,7)
      YPR(1)=YMIN
      DO 90 KN=1,9
   90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
      YPR(11)=YMAX
      WRITE(6,8) (YPR(IP), IP=1,11)
      RETURN
      END
```

```
      SUBROUTINE POLRBW(PR,N,M,PLT,PNT,TERM,X,DI,D,P,B)
***********************************************************************
*   SAMPLE MAIN PROGRAM FOR POLYNOMIAL REGRESSION - POLRG             *
*                                                                     *
*   PURPOSE                                                           *
*      (1) READ THE PROBLEM PARAMETER CARD FOR A POLYNOMIAL REGRES-   *
*      SION, (2) CALL SUBROUTINES TO PERFORM THE ANALYSIS, (3)        *
*      PRINT THE REGRESSION COEFFICIENTS AND ANALYSIS OF VARIANCE     *
*      TABLE FOR POLYNOMIALS OF SUCCESSIVELY INCREASING DEGREES,      *
*      AND (4) OPTIONALLY PRINT THE TABLE OF RESIDUALS AND A PLOT     *
*      OF Y VALUES AND Y ESTIMATES.                                   *
*                                                                     *
*   REMARKS                                                           *
*      THE NUMBER OF OBSERVATIONS, N, MUST BE GREATER THAN M+1,       *
*      WHERE M IS THE HIGHEST DEGREE POLYNOMIAL SPECIFIED.            *
*      IF THERE IS NO REDUCTION IN THE RESIDUAL SUM OF SQUARES        *
*      BETWEEN TWO SUCCESSIVE DEGREES OF THE POLYNOMIALS, THE         *
*      PROGRAM TERMINATES THE PROBLEM BEFORE COMPLETING THE ANALY-    *
*      SIS FOR THE HIGHEST DEGREE POLYNOMIAL SPECIFIED.               *
*                                                                     *
*   SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     *
*      GDATA                                                          *
*      ORDER                                                          *
*      MINV                                                           *
*      MULTR                                                          *
*      PLOT  (A SPECIAL PLOT SUBROUTINE PROVIDED FOR THE SAMPLE       *
*             PROGRAM.)                                               *
*                                                                     *
*   METHOD                                                            *
*      REFER TO B. OSTLE, "STATISTICS IN RESEARCH", THE IOWA STATE    *
*      COLLEGE PRESS, 1954, CHAPTER 6.                                *
*                                                                     *
***********************************************************************
      REAL PR(4)
      LOGICAL PLT,PNT,TERM
C
C   THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO THE
C   PRODUCT OF N*(M+1), WHERE N IS THE NUMBER OF OBSERVATIONS AND M
C   IS THE HIGHEST DEGREE POLYNOMIAL SPECIFIED.
C
      DIMENSION X(1)
C
C   THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO THE
C   PRODUCT OF M*M.
C
      DIMENSION DI(1)
C
C   THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO
C   (M+2)*(M+1)/2.
C
      DIMENSION D(1)
C
C   THE FOLLOWING DIMENSIONS MUST BE GREATER THAN OR EQUAL TO M.
C
      DIMENSION B(1),E(30),SB(30),T(30)
C
C   THE FOLLOWING DIMENSIONS MUST BE GREATER THAN OR EQUAL TO (M+1).
C
      DIMENSION XBAR(31),STD(31),COE(31),SUMSQ(31),ISAVE(31)
C
C   THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO 10.
C
      DIMENSION ANS(35)
C
C   THE FOLLOWING DIMENSION WILL BE USED IF THE PLOT OF OBSERVED DATA
C   AND ESTIMATES IS DESIRED.  THE SIZE OF THE DIMENSION, IN THIS
C   CASE, MUST BE GREATER THAN OR EQUAL TO N*3.  OTHERWISE, THE SIZE
C   OF DIMENSION MAY BE SET TO 1.
C
      DIMENSION P(1)
C
C......................................................................
C
C   IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C   C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C   STATEMENT WHICH FOLLOWS.
C
      DOUBLE PRECISION X,XBAR,STD,D,SUMSQ,DI,E,B,SB,T,ANS,DET,COE,PR,
     +P,ISAVE
C
C   THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C   APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C   ROUTINE.
C
C......................................................................
```

```
C
C  PR....PROBLEM NUMBER (MAY BE ALPHAMERIC)
C  N.....NUMBER OF OBSERVATIONS
C  M.....HIGHEST DEGREE POLYNOMIAL SPECIFIED
C  PLT...CODE FOR PLOTTING
C  PNT...CODE FOR PRINTING
C
C  PRINT PROBLEM NUMBER AND N.
C
      WRITE(6,3) PR
      WRITE(6,4) N
    3 FORMAT(27H1POLYNOMIAL REGRESSION......,4A10)
    4 FORMAT(23H NUMBER OF OBSERVATIONS,I6)
      WRITE (6,17)
C
C  PRINT OUT INPUT DATA
C
      WRITE (6,24)
   24 FORMAT (//* OBSERVATION NO.*,5X,* X VALUE*,7X,* Y VALUE*/)
      DO 120 K=1,N
      KK=M*N+K
  120 WRITE (6,25) K,X(K),X(KK)
   25 FORMAT (1X,I9,F18.5,F14.5)
      WRITE (6,17)
C
C  GENERATE INDEPENDENT VARIABLES AND COMPUTE THE MEANS,
C  STANDARD DEVIATIONS AND CORRELATION CEOFFICIENTS.
C
      CALL GDATA (N,M,X,XBAR,STD,D,SUMSQ)
      MM=M+1
      SUM=0.0
      NT=N-1
C
C  BEGIN THE LOOP TO COMPUTE POLYNOMIALS.
C
      DO 200 I=1,M
      ISAVE(I)=I
C
C  FORM SUBSET OF CORRELATION COEFFICIENT MATRIX
C
      CALL ORDER (MM,D,MM,I,ISAVE,DI,E)
C
C  INVERT THE SUBMATRIX OF CORRELATION COEFFICIENTS
C
      CALL MINV (DI,I,DET,B,T)
      CALL MULTR (N,I,XBAR,STD,SUMSQ,DI,E,ISAVE,B,SB,T,ANS)
C
C  PRINT THE RESULT OF CALCULATION
C
      WRITE(6,5)I
    5 FORMAT(//32H0POLYNOMIAL REGRESSION OF DEGREE,I3)
      SUMIP=ANS(4)-SUM
      LA=I
      IF(SUMIP) 140, 140, 150
  140 WRITE(6,13)
   13 FORMAT(17H    NO IMPROVEMENT)
      WRITE (6,17)
      IF(TERM) RETURN
  150 SUM=ANS(4)
      NI=ANS(8)
      WRITE(6,6)ANS(1)
      WRITE (6,7) (B(J),J=1,I)
      IF(.NOT.PNT)GO TO 151
      WRITE (6,8) I
      WRITE (6,9)
      WRITE (6,10) I,ANS(4),ANS(6),ANS(10),SUMIP
      WRITE (6,11) NI,ANS(7),ANS(9)
      WRITE (6,12) NT,SUMSQ(MM)
    6 FORMAT(12H   INTERCEPT,1PE18.6)
    7 FORMAT(/26H   REGRESSION COEFFICIENTS/(1H ,1P6E18.6))
    8 FORMAT(/1X,24X,24HANALYSIS OF VARIANCE FOR,I4,19H  DEGREE POLYNOMI
     +AL)
    9 FORMAT(1H ,5X,19HSOURCE OF VARIATION,7X,9HDEGREE OF,7X,6HSUM OF,9X
     +,4HMEAN,10X,1HF,9X,20HIMPROVEMENT IN TERMS/33X,7HFREEDOM,8X,7HSQUA
     +RES,7X,6HSQUARE,7X,5HVALUE,8X,17HOF SUM OF SQUARES)
   10 FORMAT(20H   DUE TO REGRESSION,12X,I6,F17.5,F14.5,F13.5,F23.9)
   11 FORMAT(32H   DEVIATION ABOUT REGRESSION   ,I6,F17.5,F14.5)
   12 FORMAT(8X,5HTOTAL,19X,I6,F17.5)
  151 WRITE (6,17)
   17 FORMAT (//,1X,115(1H*))
C
C  SAVE COEFFICIENTS FOR CALCULATION OF Y ESTIMATES
C
      COE(1)=ANS(1)
      DO 160 J=1,I
  160 COE(J+1)=B(J)
```

```
C
C  TEST WHETHER PLOT IS DESIRED
C
       IF(.NOT.PLT)GO TO 200
C
C  CALCULATE ESTIMATES
C
   220 NP3=N+N
       DO 230 K=1,N
       NP3=NP3+1
       P(NP3)=COE(1)
       L=K
       DO 230 J=1,LA
       P(NP3)=P(NP3)+X(L)*COE(J+1)
   230 L=L+N
C
C  COPY OBSERVED DATA
C
       N2=N
       L=N*M
       DO 240 K=1,N
       P(K)=X(K)
       N2=N2+1
       L=L+1
   240 P(N2)=X(L)
       IF(.NOT.PNT)GO TO 255
C
C  PRINT TABLE OF RESIDUALS
C
       WRITE (6,14)
    14 FORMAT(1H // 2X,18HTABLE OF RESIDUALS//16H OBSERVATION NO.,5X,7HX
      +VALUE,7X,7HY VALUE,7X,10HY ESTIMATE,7X,8HRESIDUAL,5X,12H% ERROR
      + Y/)
       NP2=N
       NP3=N+N
       DO 250 K=1,N
       NP2=NP2+1
       NP3=NP3+1
       RESID=P(NP2)-P(NP3)
       XXX=0
       IF(DABS(P(NP2)).GT.1E+50) GO TO 250
       XXX=RESID/P(NP2)*100.
   250 WRITE (6,15) K,P(K),P(NP2),P(NP3),RESID,XXX
    15 FORMAT(1H ,3X,I6,F18.5,F14.5,F17.5,F15.5,F13.2)
   255 WRITE (6,17)
   919 FORMAT (//)
       WRITE (6,919)
       CALL PLOT (LA,P,N,3,0,1)
       WRITE (6,17)
       IF(TERM.AND.SUMIP.LE.0.)RETURN
   200 CONTINUE
   260 RETURN
       END
```

```
      SUBROUTINE PREDIST (DATA,LDIM,LSAMPL,N,C,L)
C**********************************************************************
C*  THIS SUBROUTINE PREDISTORTS A GIVEN SEQUENCE OF DIGITAL           *
C*  SIGNAL BY PASSING IT THROUGH A TRANSVERSAL FILTER OF N-TAPS.      *
C**********************************************************************
      DIMENSION DATA(LDIM),C(N),X(100),Y(100)
      LSAMPL2=2*LSAMPL
      L2=N+L-1
      DO 11 I=1,100
   11    X(I)=0.
      DO 22 I=1,L2
   22    X(I)=DATA(I*LSAMPL2-1)
      DO 44 I=1,L2
         Y(I)=0.
         K=I
         DO 33 J=1,N
            IF(J.GT.I) GO TO 44
            Y(I)=Y(I)+X(K)*C(J)
   33       K=K-1
   44 CONTINUE
      DO 66 I=1,L2
         K=LSAMPL2*(I-1)
         DO 55 J=1,LSAMPL2,2
   55       DATA(J+K)=Y(I)
   66 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RCOSMOD (SBANDW,BIRATE,TF,LDIM)
***********************************************************************
*  THIS SUBROUTINE COMPUTES THE TRANSFER FUNCTION OF A RAISED COSINE *
*  FILTER (-6.0 DB AT FREQUENCY=1/2T). THE TRANSFER FUNCTION IS ALSO *
*  MODIFIED BY (WT/2)/SIN(WT/2) SO THAT THIS FILTER RESULTS IN ZERO  *
*  INTER-SYMBOL-INTERFERENCE FOR ANY ALFA (0. TO 1.0).               *
***********************************************************************
      DIMENSION TF(LDIM)
      READ *,ALFA
      PRINT 1,ALFA
      PI=4.*ATAN2(1.,1.)
      NH=LDIM/2
      NQ=NH/2
      DW=SBANDW/FLOAT(NH)
      W=0.0
      WM=BIRATE*(1.-ALFA)/2.
      WP=BIRATE*(1.+ALFA)/2.
      DO 11 I=1,NQ
         IF(W.EQ.0.0) GO TO 10
         WW=PI*W/BIRATE
         WWW=WW/2./ALFA-PI*(1.-ALFA)/4./ALFA
         IF(W.LE.WM) F=WW/SIN(WW)
         IF(W.GT.WM.AND.W.LE.WP) F=WW/SIN(WW)*COS(WWW)**2
         IF(W.GT.WP) F=0.0
         GO TO 20
   10    F=1.0
   20    FA=-(I-1)*PI/2.
         X=F*COS(FA)
         Y=F*SIN(FA)
         I2=I*2
         I1=I2-1
         TF(I1)=X
         TF(I2)=Y
         I2=2*(NH-I+2)
         I1=I2-1
         IF(I1.GT.LDIM.OR.I2.GT.LDIM) GO TO 3
         TF(I1)=X
         TF(I2)=-Y
    3    W=W+DW
   11 CONTINUE
    1 FORMAT(1H1,4X,*A RAISED COSINE FILTER IS USED. ITS EXCESS *
     +*NYQUIST BANDWIDTH IS *,F4.2//)
      RETURN
      END
```

```
      SUBROUTINE SCUTF6 (SBANDW,BIRATE,TF,LDIM)
************************************************************
*  THIS SUBROUTINE COMPUTES THE TRANSFER FUNCTION OF A SHARP CUT *
*  OFF FILTER WITH -6.0 DB AT BIRATE/2.                         *
************************************************************
      DIMENSION TF(LDIM)
      READ,*,ALFA
      PRINT 1,ALFA
      PI=4.*ATAN2(1.,1.)
      NH=LDIM/2
      NQ=NH/2
      DW=SBANDW/FLOAT(NH)
      W=0.0
      WM=BIRATE*(1.-ALFA)/2.
      WP=BIRATE*(1.+ALFA)/2.
      DO 11 I=1,NQ
         IF(W.EQ.0.) GO TO 10
         WW=PI*W/BIRATE
         WWW=WW/2./ALFA-PI*(1.-ALFA)/4./ALFA
         IF(W.LE.WM) F=1.0
         IF(W.GT.WM.AND.W.LE.WP) F=COS(WWW)**2
         IF(W.GT.WP) F=0.
         GO TO 20
 10      F=1.0
 20      FA=-(I-1.)*PI/2.
         X=F*COS(FA)
         Y=F*SIN(FA)
         I2=I*2
         I1=I2-1
         TF(I1)=X
         TF(I2)=Y
         I2=2*(NH-I+2)
         I1=I2-1
         IF(I1.GT.LDIM.OR.I2.GT.LDIM) GO TO 3
         TF(I1)=X
         TF(I2)=-Y
  3      W=W+DW
 11   CONTINUE
  1   FORMAT(1H1,4X,*A SHARP CUT-OFF CILTER IS USED. ITS EXCESS *,
     +*NYQUIST BANDWIDTH IS *,F4.2//)
      RETURN
      END
```

```
      SUBROUTINE THEOFIL(SBANDW,FBANDW,TF,LDIM)
*****************************************************************
*  THIS SUBROUTINE GENERATES THE TRANSFER FUNCTION OF A CHEBYSCHEFF *
*  OR A BUTTERWORTH LPF     WHOSE CHARACTERISTIC IS DETERMINED BY   *
*  COEFFICIENTS OF THEIR POLYNOMIALS SUPPLIED TO ARRAY 'C'.         *
*  THE DIMENSIONS OF WW AND C ARE SET FOR A MAX.  FILTER ORDER OF 8.*
*  THE VALUES OF N (ORDER+1) AND C (COEFFICIENTS ) MUST BE GIVEN AS *
*  INPUT. DIMENSION OF WW MUST BE LARGER THAN THAT OF C. FOR A      *
*  HIGHER ORDER, CHANGE DIMENSIONS OF ARRAYS 'C' & 'WW'.            *
*****************************************************************
      DIMENSION C(9),WW(10),TF(LDIM),A(4)
      READ 1,(A(I),I=1,4)
      READ *,N,(C(I),I=1,N)
      PRINT 2,(A(I),I=1,4)
      PRINT 3,N,(C(I),I=1,N)
      NH=LDIM/2
      NQ=LDIM/4
      DW=SBANDW/FLOAT(NH)/FBANDW*2.
      DO 44 I=2,NQ
         W=DW*FLOAT(I-1)
         WS=-W*W
         SW=1.
         DO 11 J=1,N,2
            J1=J+1
            WW(J)=SW
            WW(J1)=SW*W
 11         SW=SW*WS
         SX=SY=0.
         DO 22 J=1,N,2
 22         SX=SX+C(J)*WW(J)
         DO 33 J=2,N,2
 33         SY=SY+C(J)*WW(J)
         SS=(SX**2+SY**2)/C(1)
         X=SX/SS
         Y=SY/SS
         I2=I*2
         I1=I2-1
         TF(I1)=X
         TF(I2)=-Y
         I2=2*(NH-I+2)
         I1=I2-1
         TF(I1)=X
 44      TF(I2)=Y
      TF(1)=1.
      TF(2)=TF(NH+1)=TF(NH+2)=0.
    1 FORMAT(4A10)
    2 FORMAT(1H1,4X,*A THEORETICAL FILTER IS USED. THE FILTER TYPE IS,*
     +/5X,4A10//)
    3 FORMAT(5X,*ITS *,I3,* POLYNOMIAL COEFFICIENTS ARE,*//
     +9(5X,F9.5))
      RETURN
      END
```

```
      SUBROUTINE TRANSF (A,SIGNAL,N,K,LDIM)
*****************************************************************
*  THIS SUBROUTINE TRANSFORMS THE REAL ARRAY 'SIGNAL' *
*  INTO A COMPLEX ARRAY 'A', AND VICE-VERSA DEPENDING *
*  UPON THE VALUE ASSIGNED TO K.                       *
*****************************************************************
      DIMENSION SIGNAL(LDIM)
      COMPLEX A(N)
      IF(K .NE. 0) GO TO 10
      DO 11 I=1,N
         II=I*2
         IR=II-1
         XR=SIGNAL(IR)
         XI=SIGNAL(II)
 11      A(I)=CMPLX(XR,XI)
      RETURN
   10 DO 22 I=1,N
         II=I*2
         IR=II-1
         SIGNAL(IR)=REAL(A(I))
 22      SIGNAL(II)=AIMAG(A(I))
      RETURN
      END
```

C.5 Compilation and Execution of Programs and Subroutines

Compilation of a program with its associated subroutines can become a time-consuming (in computer System Resource Units) procedure. In order to overcome this wastage of resources; it is a good practice to compile all common subroutines only once and store the compiled versions in a library file. This file could be accessed during the execution of programs. This procedure needs a careful organization of one's permanent file library. It also prevents from wasting the memory space allocated to the user when writing the same subroutine in several programs.

The programs were run on the Time Sharing System, in BATCH mode, of CDC Cyber 172 computer system at Concordia University Computer Centre. In BATCH mode the user can compile and execute the programs at a video or printer terminal with the option of printing long lists of results at a remote high speed printer through the disposal technique (i.e., using the PRINT command).

<u>Compilation of Subroutine Library.</u> As explained in the earlier section, following is the procedure to generate a library file in BATCH mode. (The slash '/' is the system response to enter a line.)

```
/REWIND,FNAME
/FTN,I=FNAME
/LIBGEN,P=MYLIB
/SAVE,MYLIB
```

When a file is compiled, unless specified, the system automatically generates a file called LGO that contain the compiled version of input file. Thus, if another input file is now to be compiled, then, LGO must either be first rewound (/rewind,lgo) or returned (/return,lgo).