DIGITAL SIMULATION OF DELTA MODULATION

by

JACK SHIGEO MATSUSHITA

B.A.Sc., University of British Columbia, 1959

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the

Department of Electrical Engineering

The University of British Columbia

November 1960

DIGITAL SIMULATION OF DELTA MODULATION

by

JACK SHIGEO MATSUSHITA
B.A.Sc., University of British Columbia, 1959

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

In the Department of
Electrical Engineering

We accept this thesis as conforming to the
standards required from candidates for the
degree of Master of Applied Science

Members of the Department
of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA
November 1960

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of ELECTRICAL ENGINEERING

The University of British Columbia,
Vancouver 8, Canada.

Date Nov. 25, 1960

ABSTRACT

This thesis describes the simulation of a digital communication system on a digital computer. Delta modulation was chosen as the system, and its mode of action is first described. Several variations of the basic system are possible. In order to get the best transmission quality, a careful choice must be made of the system and of its design parameters. Conventional methods of finding these optimum parameters have difficulties which digital simulation can circumvent.

The programming of the ALWAC III-E computer for this task is described. Difficulties were encountered due to the modest speed of the computer. The simulation experiments yielded many results of interest concerning the operation of both simple and complex delta modulation systems with different design parameters, and allowed an optimum system to be designed. Where it is possible to compare results with previous experimental work, the agreement is good.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENT

Acknowledgement is gratefully given to the Defence Research Board, Department of National Defence, for a research assistantship and for their sponsorship of this project (Grant No. 2803-19), and also to the National Research Council for their assistance under block term grant BT 68. The author also wishes to thank the British Columbia Telephone Company for the scholarship awarded him in 1959.

The author is indebted to Professor F. K. Bowers for his guidance throughout the project, and to Dr. F. Noakes and other members of the Electrical Engineering Department for their help. Special thanks go to Mr. H. Dempster of the U.B.C. Computing Centre for his aid in the programming.

DIGITAL SIMULATION OF DELTA MODULATION

1. INTRODUCTION

The work reported in this thesis started as a continuation of P. J. deFaye's project[1]. He had constructed transistorized equipment for transmitting speech in the form of binary pulses, using "Delta Modulation". The objective of the present work was to examine variations of encoding schemes to find one able to transmit the signal with the least possible amount of distortion.

After considering various methods of performing this optimization, it was decided to try an indirect approach. Instead of building actual equipment using various schemes, the ALWAC III-E computer was programmed to compute what the equipment would do if it were built. Due to the modest speed of this computer, this approach posed some challenging problems. However, the method was a fruitful one, and gave insight both into the characteristics of delta modulation systems and into the capabilities of this type of computer for simulating digital communication systems generally.

The simulation process was therefore pursued much further than had been originally intended, and the compilation of different computer programs forms a large part of this thesis. No effort was made to apply the results of these simulation trials to the actual equipment.

2. THE DELTA MODULATION SYSTEM

2.1 Delta Modulation

Delta Modulation[2] is a method of communicating information, such as speech, by means of a series of binary pulses. It may be regarded as a simple form of pulse code modulation, requiring only a single digit code. In this system, the signal to be transmitted is first sampled. The sampling rate used is much higher than the theoretically required minimum of 2f, where f is the upper cut-off frequency of the input signal. At each sampling instant, the amplitude of the input signal is compared with the amplitude of a waveform similar to the signal reconstructed at the receiver from previously transmitted information, and the polarity of the difference is transmitted by a pulse. Thus a single digit binary code is sufficient. The simplicity of the required code suggests the possibility of inexpensive equipment, simpler than that needed for other forms of pulse code modulation, but still retaining their desirable characteristics: the pulse train to be transmitted is a rugged signal and can, in principle, be transmitted, switched, or stored with no deterioration whatever.

Various investigators[1,2,3] have reported on slightly differing delta modulation systems, but they all follow the basic circuit illustrated in Figure 2.1. In the coder, the pulse generator is triggered by the clock pulses, emitting either a positive or a negative pulse depending on the polarity at its input. These pulses form the output $e_o(t)$. The output is also introduced into a feedback loop containing a network Q equivalent

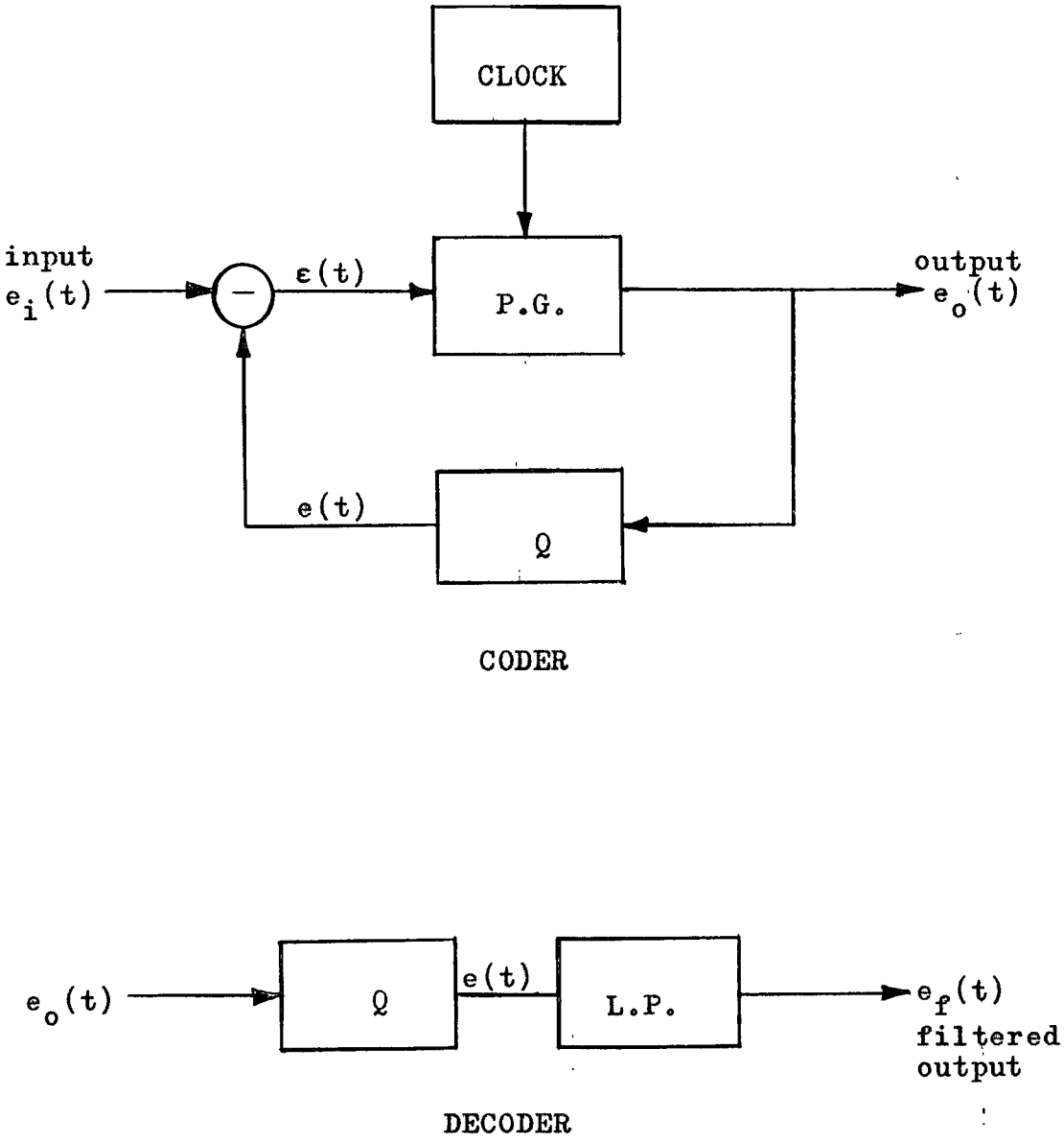


Figure 2.1 Delta Modulation System

to the network in the receiver. The feedback waveform e(t), which will be similar to the output of the receiver just before the low pass filter, is compared to the input signal $e_i(t)$, and the difference forms the error signal $\varepsilon(t)$, which determines the polarity at the input to the pulse generator. This feedback action thus tends to make the receiver output follow the coder input.

2.2 Variations of Delta Modulation Systems

The variations in the several working systems lie in the network Q used in the feedback loop and in the receiver network. The most elementary loop is simply an integrator, usually a capacitative network. As illustrated in Figure 2.2, at each sampling instant a positive or a negative pulse of height h will increase or decrease the step waveform e(t) produced at the integrator by one unit step of height h. This step wave will tend to follow the input signal quite closely. The difference, or "quantizing" error will be a function of the sampling frequency and of the step height h.

In a practical system, a true integrator is hard to achieve because of the finite time constant associated with capacitors, but with a sampling frequency in the order of 100 ks/c, a good approximation to a perfect integration could be obtained. However, a finite time constant is not necessarily a handicap, it can also be used to good advantage. The American Army Signal Corps has developed a system which they call Exponential Delta Modulation[3]. A finite time constant RC network is used instead of an integrator, resulting in a waveform with an effectively

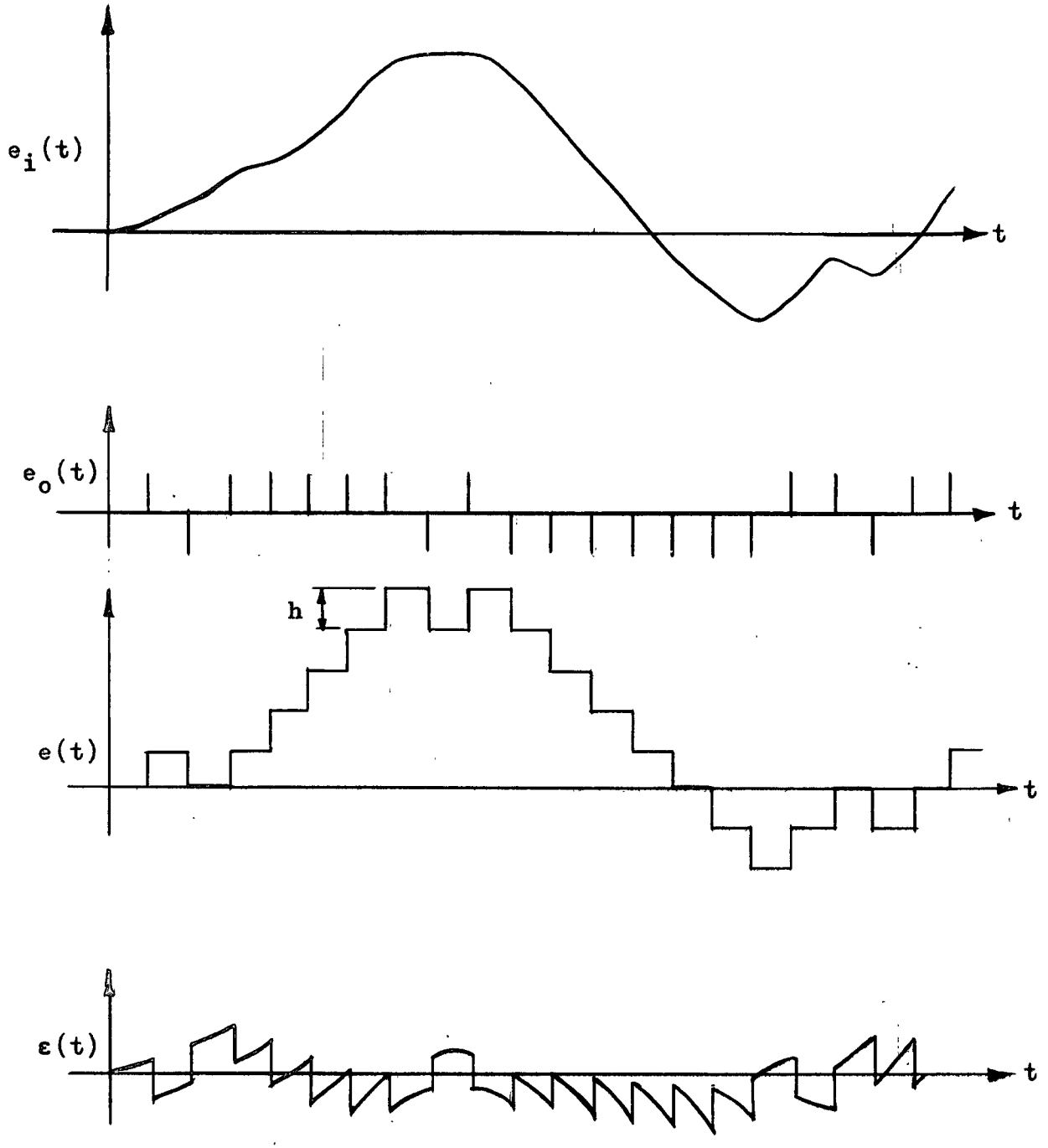


Figure 2.2 Typical Waveforms of Simple Delta Modulation

non-uniform step size, as opposed to the equal increments due to true integration. This system has an advantage in that it is now possible to transmit d-c levels. They are transmitted by pulse patterns varying from one of all pulses, to one of all spaces.

Several more complex feedback loops have been tried in the coder, in the hope that this will produce a closer resemblance between the input signal and the reproduced signal. One of the most effective ways to improve on the simple coder is to use multiple integration networks in the feedback loop. In his original paper on delta modulation, de Jager[2] describes also a second order system using double integration in the feedback loop. After passing through two stages of integration, a single pulse will produce a change in the slope of e(t) of $\pm\Delta$. Unfortunately, such a system has a tendency to oscillate, as is illustrated in Figure 2.3. This shows the behaviour of a second order system following a sudden change in the input signal $e_i(t)$. The oscillation thus produced differs from those normally found in feedback loops in that its period can have many different values, depending on initial conditions.

To overcome this instability, de Jager used what he calls "prediction". The approximating curve e(t), is built up of straight lines so that if no change of slope occurs, the value which would be reached after a time interval τ can be predicted. This extrapolated value of the approximating curve is compared to the value of the input signal at that time, and the decision as to the polarity of the next output pulse is made. (Figure 2.4).

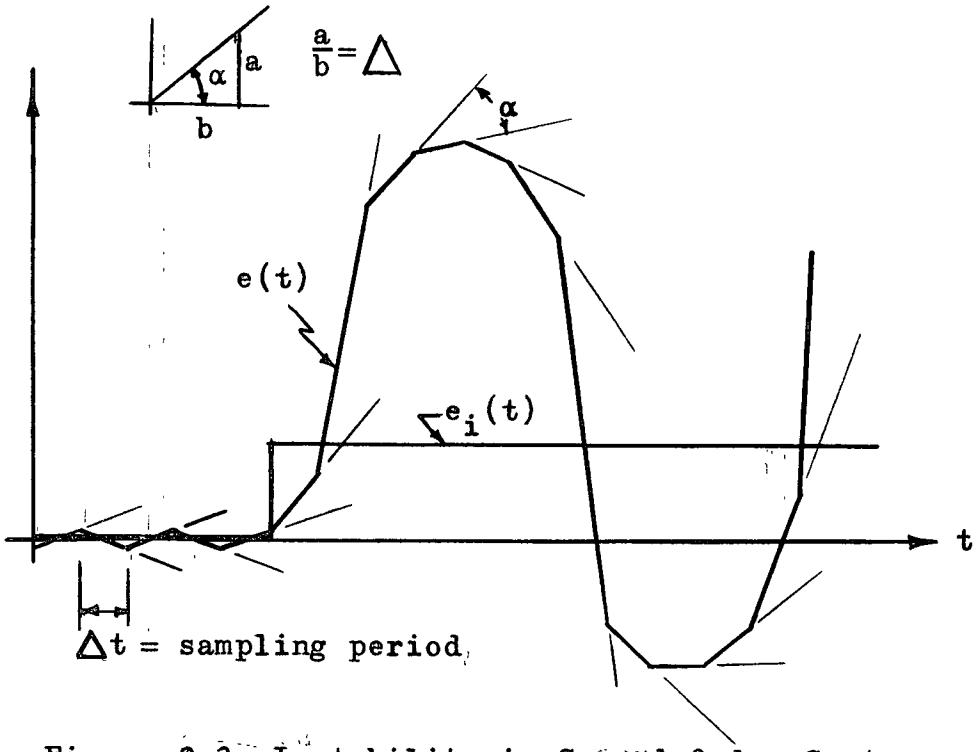


Figure 2.3 Instability in Second Order Systems

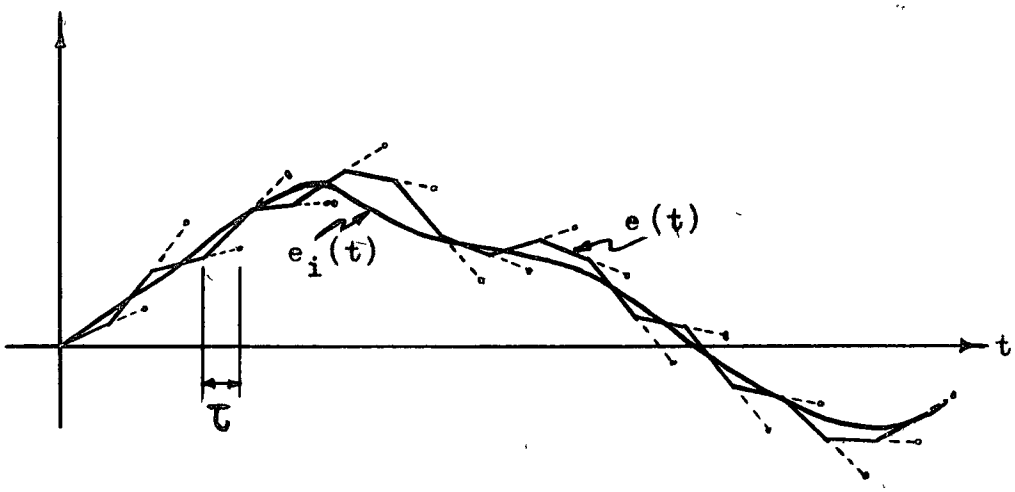


Figure 2.4 Double Integration with Prediction

The electrical system required for this prediction must make the output signal e_2 at the second integrator at time $t + \tau$ equal to the input signal e_i at time t. A network must be constructed which generates the extrapolated value of e_2 such that

$$e(t) = e_2(t + \tau)$$

This network, if placed in the feedback loop, will provide the necessary predicting action. Figure 2.5 shows such a network.

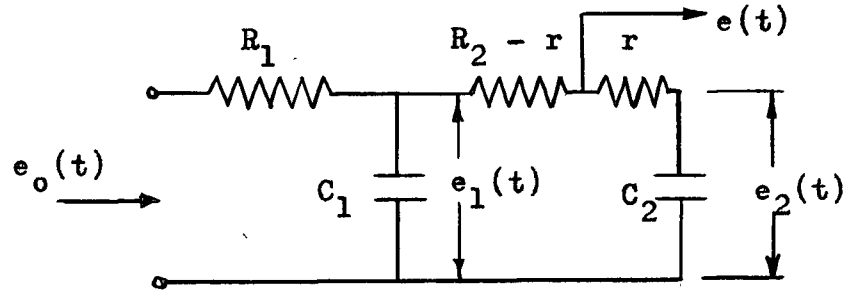


Figure 2.5 Network for Double Integration with Prediction.

Now,

$$e_2(t + \tau) = e_2(t) + \tau \frac{de_2}{dt}$$

and

$$\frac{de_2}{dt} = \frac{i}{C_2}$$

The desired feedback voltage e(t) can be obtained by adding $\frac{\tau i}{C_2}$ to the voltage on C_2. Examination of Figure 2.5 shows that this network will provide the necessary voltage, with

$$\tau = rC_2 ,$$

τ being small compared to R_1C_1 and R_2C_2.

The network of Figure 2.5 may be also be regarded in a

different light. Since $e_1(t)$ and $e_2(t)$ are respectively the result of one and of two stages of integration, the output $e(t)$ can be written as a linear combination of these two signals, where the proportion of second-order integration may be judiciously adjusted to obtain the closest resemblance of signals $e_0(t)$ and $e(t)$ consistent with adequate stability.

It may then be possible to improve the performance of the system still further by a small admixture of third-order integrations. There is hardly any limit to the possible types of feedback loop that might be tried, some perhaps designed to fit a particular type of input signal.

Very little study has been made of such complex delta modulation systems, but it is clear there is a wide scope for trial and error, and a great need exists for methods of evaluating the performance of such systems quantitatively, and speedily. The development of one such method is the subject of this thesis.

3. SIGNAL TO NOISE CONSIDERATIONS

3.1 Overload and Quantizing Noise

In transmission, pulses can in theory be regenerated so that no error is introduced, but even then the reproduced signal will not be identical with the input signal. If the input signal is too large, very great discrepancies appear. The overload amplitude is an important characteristic of any communications system. However, even for smaller inputs there will be differences or distortions due to the finite size of the step height h. Such distortions are called "quantizing noise" and every effort is made to keep this type of noise to a minimum. In particular, it is desired to have the largest possible "signal-to-noise ratio", that is, the ratio of the overload signal power to the average quantizing noise power.

A particularily noticeable form of quantizing noise occurs in some delta modulation systems when the input is very small, with amplitude less than h, as illustrated in Figure 3.1. The system then transmits the zero level signal and a "threshold effect" is observed.

Both ordinary quantizing noise and the threshold effect are reduced if the step height is reduced. This, however, also decreases the overload point, and does not alter the overall signal-to-noise ratio or the dynamic range of the system. Considerable improvement results if the sampling frequency f is increased: the overload point increases, while quantizing noise is decreased. However, the greater sampling frequency requires a wider transmission bandwidth, and the real test of the performance of a

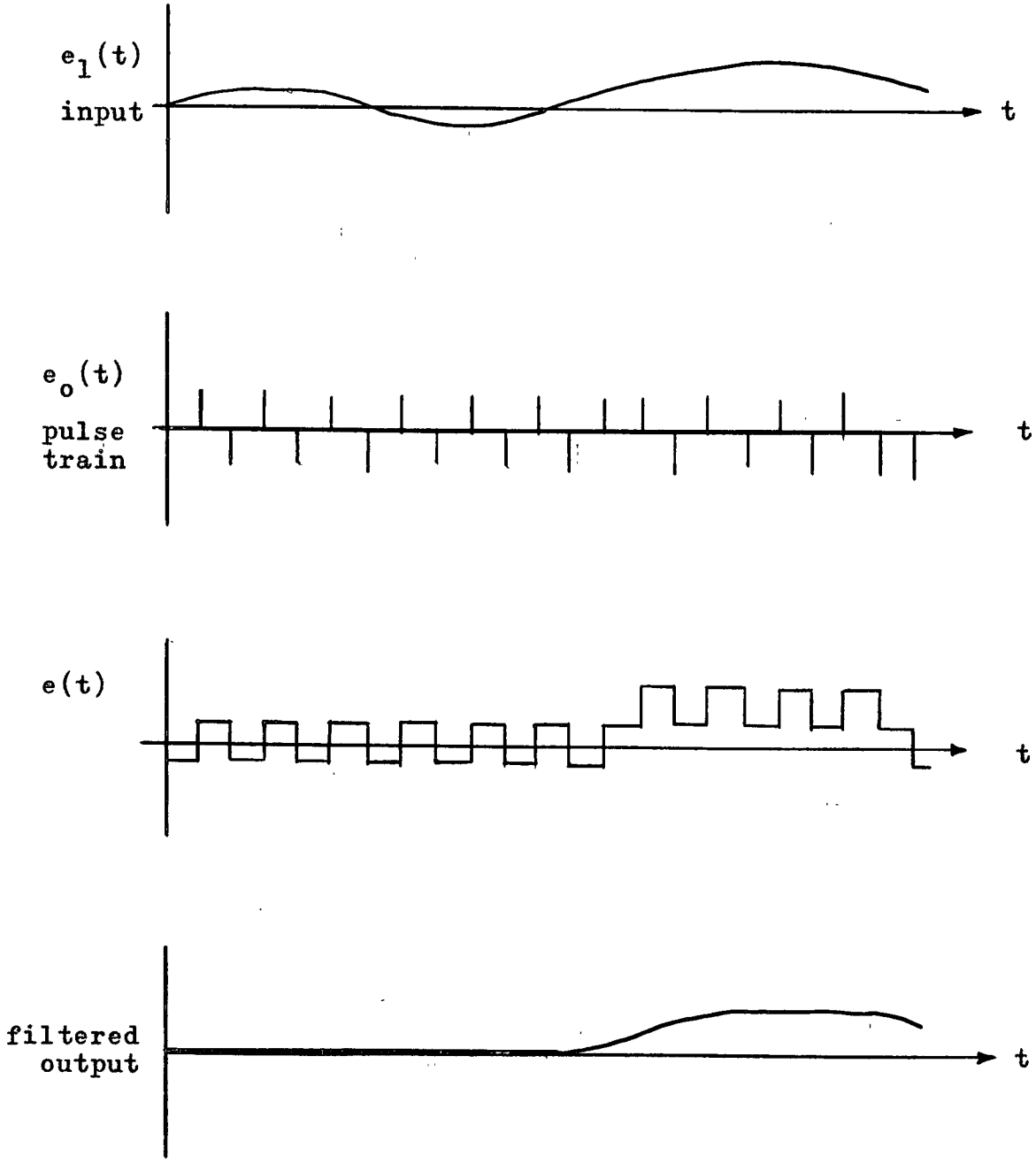


Figure 3.1 Threshold Effect

particular pulse communication system should be the quantizing noise it produces with a given overload point and a given pulse frequency.

3.2 Standardizing Overload Characteristics

When the signal-to-noise ratios of several variants of delta modulation systems are to be compared, it is generally necessary to find for each system both the overload characteristics and the quantizing noise. However, there is a technique for ensuring that all these systems have the same overload characteristics, and this will now be described.

If a linear network N were added to the receiver, this can have a drastic effect on the frequency response of the whole system, but at any frequency, it will have the same effect on the maximum amplitude that can be received before overload occurs as it has on the quantizing noise at that frequency. Hence N does not alter the signal-to-noise ratio of the received signal. To undo the damage done to the frequency response of the entire system by this network, an inverse network N' can be placed in front of the transmitter, such that the combination of N and N' has a flat frequency response.

By a suitable choice of N-N' networks, it is possible to manipulate the overload characteristics of any variant system until it is the same as that of the standard system, say simple delta modulation, without altering the signal-to-noise ratio of the variant system. To optimize the system, then, it is necessary to examine only the quantizing noise, and to look for a system which will make this a minimum.

Figure 3.2 illustrates the procedure for a double integration delta modulation system. One of the integrators in the receiver must be omitted to give it the same overload spectrum as single integration, so that in this case N is a differentiator network, and N' is an integrator. After some reorganization of the block-diagram (Figure 3.2d), it is seen that this variant can be regarded as having the same feedback loop as simple delta modulation, but differing from it in the manner in which the decision between positive and negative pulses is made. In simple delta modulation, the decision function d(t) is $\varepsilon(t)$, the error signal, whereas in the double integration system, d(t) is $\int \varepsilon(t)dt$.

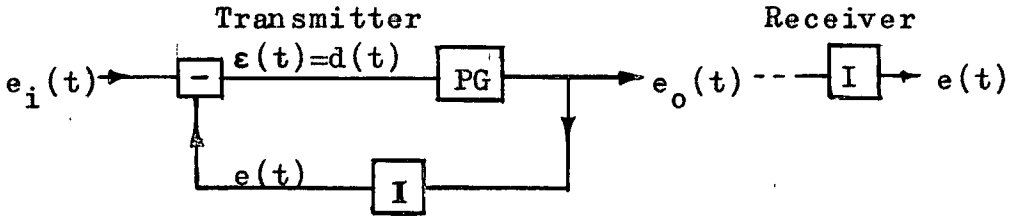
All other variations of the feedback loop, using linear elements, can be similarly re-drawn as a circuit with a loop containing one integrator, but with more complicated decision functions. This is the viewpoint adopted for the simulation experiments, and attention need then be directed only at the quantizing noise accompanying assorted input signals.

3.3 The Signal-to-Noise Ratio Determinations

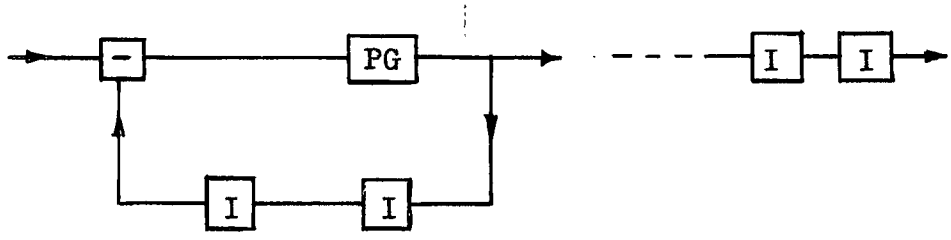
Signal-to-noise ratios are normally found by three possible methods: by calculation, by measurement, and by listening tests. Each method will be discussed in turn, and each will be found to have its particular difficulties.

3.3.1 Calculation of Signal-to-Noise Ratio

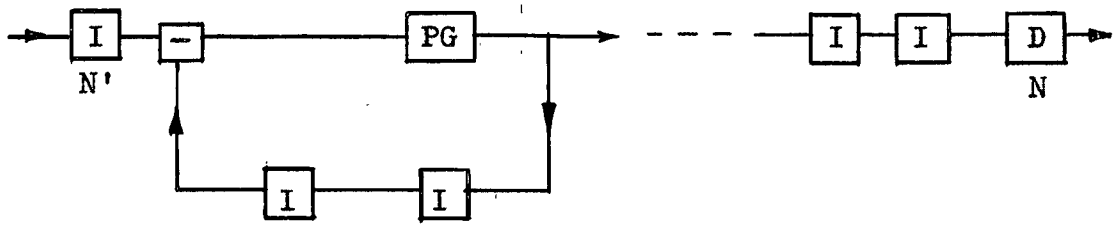
The overload point for simple delta modulation is easily calculated. As can be seen from Figure 3.3, the maximum slope the step wave e(t) can follow is given by



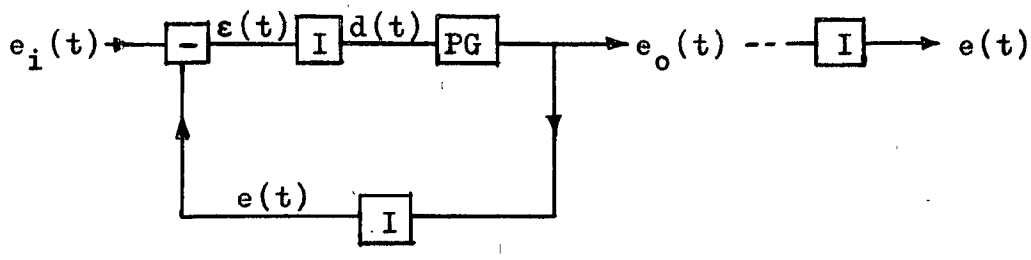
a) The "standard" system: Simple Delta Modulation.
(I = Integrator, PG = Pulse Generator, with pulse polarity depending on the polarity of the decision function d(t).)



b) Double Integration Delta Modulation.
(Overloads when second derivative of input signal is too large.)



c) Networks N and N' added to b) to get the same overload spectrum as system (a): overloads when first derivative is excessive.
(D = Differentiator. Addition of N and N' does not change the signal-to-noise ratio at any frequency.)



d) Re-organisation of (c) to show similarity to (a).
(Note that the difference between two integrals is the same as the integral of the difference signal.)

Figure 3.2 Standardizing the Overload Characteristics of a Double Integration Delta Modulation System.

$$s_{max} = \frac{h}{1/f_s}$$

Thus the maximum amplitude a sine wave of frequency f can have, is given by

$$A_{max} = \frac{hf_s}{2\pi f} \quad \text{or} \quad S = \left(\frac{hf_s}{2\pi f}\right)^2$$

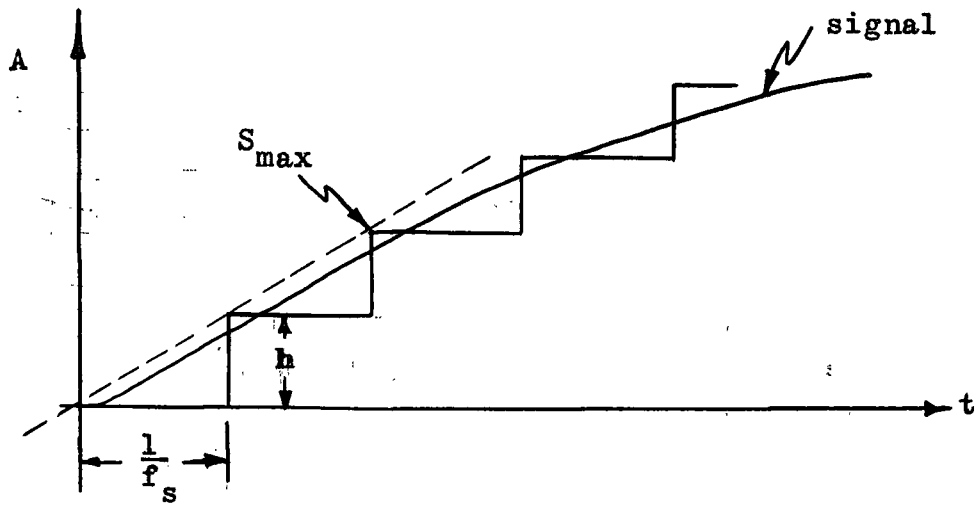


Figure 3.3 Slope Overload

At the receiver the decoded signal is usually passed through a low-pass filter which has an upper cut-off frequency f_o, the cut-off frequency of the input signal. Hence only the low frequency components, those less than f_o, are of interest in studying the noise. The quantizing noise arises from the low frequency components in the difference signal $\varepsilon(t) = e_i(t) - e_o(t)$. For most signals which are not too large or too small, the amplitudes of successive segments of ε are randomly distributed over the range $\pm h$. (cf. Fig. 2.2). On this assumption, de Jager has obtained an expression for the signal-to-noise ratio. For a sinewave of frequency f, with step height h, and sampling frequency f_s, the

ratio is given by

$$\frac{S}{N} = c_1 \frac{f_s^{\frac{3}{2}}}{f \cdot f_o^{\frac{1}{2}}}$$

The constant c_1 was found to be approximately 0.20.

For a system with double integration, the corresponding expression turns out to be

$$\frac{S}{N} = c_2 \frac{f_s^{\frac{5}{2}}}{f \cdot f_o^{\frac{1}{2}}}$$

Here the constant c_2 was found by numerical methods to be 0.026.

For a mixture of higher orders of integration, it is not possible to calculate their effect on the noise. Similarly, computation of noise for signals near the threshold is not possible.

3.3.2 Experimental Measurements of Signal-to-Noise Ratios

There is in principle no difficulty in such measurements. A reliable delta coder must first be constructed, and it must then be supplied with a variety of input signals. The quantizing noise is most conveniently measured by examining the error wave-form $\varepsilon(t)$ present in the coder, and filtering out all the irrelevant high frequency components.

Two difficulties arise:

1) It is very hard to be sure that the noise is not influenced by small imperfections in the apparatus, and
2) The noise will not be completely gaussian and uncorrelated with the signal, and its subjective effect on transmission quality may be very different from that of the same amount of gaussian noise.

3.3.3 Listening Tests

The only real criterion of the quality of a transmission system is to have several observers listen to the transmitted signals, and compare them with the sound of the original signal plus various amounts of added random noise. A noise figure can be arrived at when an amount of added noise is found which is just as objectionable as the quantizing noise accompanying the transmitted signal.

Such tests require a long time, many patient observers, and a large variety of input signals. They are subject to human errors - observers become accustomed to certain types of distortion - and also to errors due to imperfection in the equipment. Although such tests are the only final criterion by which a system is to be judged, they are not practical for the determination of the optimum value of a large number of parameters. For this, a faster method of evaluation is needed, preferably one that does not depend on perfection of the apparatus.

Both signal-to-noise ratio measurements and listening tests are slow and unreliable methods of determining the value of a system. In both cases, a working model must first be constructed, often a slow and costly process. Experimental imperfections of the model can invalidate results using either method. A speedier and more reliable method of evaluation is required. One such method, used in this thesis, is simulation of the system on a digital computer[4]. With the increasing availability of general-purpose computers, digital simulation becomes more and more practical. Digital simulation has an advantage in its high speed and low

costs, and more important, in the case of sampled data systems, in its reliability. For in this case, the simulation can be made exact.

4. DIGITAL SIMULATION OF A DELTA MODULATOR

4.1 Program Requirements

The arrangement required for digital simulation is illustrated in Figure 4.1.

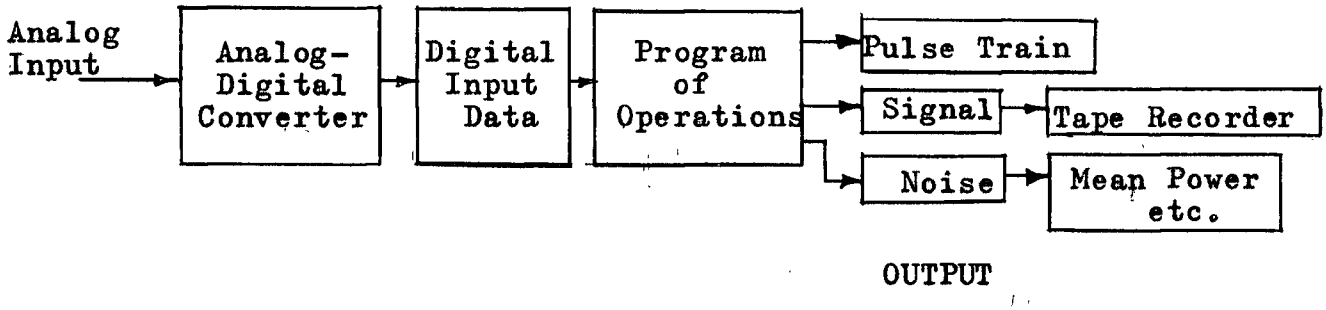


Figure 4.1 Digital Simulation Arrangement

The analog signal is first processed and converted to digital form. The digital data are then stored in some convenient form, such as on tape, and fed into the computer, where the actual operations are carried out. The results of these operations form the output, which will be in digital form. This is then converted back to an analog signal. Various other forms of output may also be desired. In the case of the delta modulator, the desired outputs are the pulse train, the signal at the receiver, the amplitudes of the quantizing noise, the mean power and other characteristics of the quantizing noise, such as its low frequency components.

The ideal computer for this purpose would have:

a) Input and output elements capable of handling whole segments of actual speech.

b) Large data storage facilities to allow a significant test.

c) A program compiler to allow easy changes in the system. A computer to fit these specifications was not available. However, this thesis will show that even on a modest sized computer useful results may be obtained. The ALWAC III-E, a medium sized, medium speed general purpose computer was programmed to optimize design parameters in a delta modulation system.

In the ALWAC III-E, inputs and outputs are handled either through a flexowriter with punched tape control, or through a high speed paper tape reader and punch. The speeds at which they operate are, for inputs, 10 and 150 characters per second respectively, and for outputs, 10 and 50 characters per second. These speeds severely limit the data handling capacity, especially at the input. Taking into account these limitations, a model delta modulator must be programmed which will allow studies of the pertinent characteristics of the system. The main points of interest will be the changes in the noise power at the output due to various changes in the feedback loop parameters. Also of interest will be a closer look at the nature of the signal-to-noise ratio, a study of its dependence on such parameters as input level.

The model must be flexible enough to allow changes in the feedback loop without extensive changes in the main program; it must yield its output, especially noise power and filtered noise power, in a form that allows rapid comparison and evaluation; and above all, it must give significant results in a short time - say ten minutes - of machine time. This is most essential, since, once the basic system is established, various changes will be made in the feedback loop, and the resulting changes in the output will be evaluated, with the aim of optimizing the whole system. As

there are many parameters to be adjusted, the time for a single run through the program must be kept to a minimum. This last requirement rules out the use of ALCOM, the automatic program compiler available in the ALWAC computer, and instead, an "optimum" program must be written.

4.2 Choosing the Input Signal

Ideally, the input should be a "typical" sample of speech or of a random signal with the same spectral characteristics as speech. Since in the computer trials it would be possible to process only about 50 milliseconds worth of input signal, it would be difficult to make sure that representative samples of such input signals are selected. Furthermore, even with so short a sample, the process of supplying to the computer values of the input at 5000 instants of time is quite tedious.

It was therefore decided to use an artificial input signal which can be generated by the computer itself, given some initial conditions. In speech, several frequency components are usually present simultaneously; the average power spectrum has a maximum below 1000 c/s, with power decreasing rapidly at higher frequencies. The final choice for an artificial input signal, determined partly by convenience in programming, was a sine wave of 500 c/s, accompanied by a fixed proportion of fourth harmonic, 20% in amplitude. In later trials, an adjustable amount of d-c was also added. The amplitude of the composite signal is specified at the beginning of each trial, and many different amplitudes were used.

Due to the regularities of this artificial signal, it was

only necessary to process about 4 milliseconds worth of input. This gave computer times of about 2 minutes per trial.

Some checks were performed to show that the performance of the system tested does not depend critically on the exact ratio of the components of the input signal, and there is good reason to hope that the performance would also be very similar for samples of actual speech.

The subroutines for sines and cosines available in the computer library were too slow, and the values of the sine waves were therefore generated as part of the program. The method used is the simultaneous solution of two differential equations, as shown in Appendix I. The values chosen for k were 2^{-3} and 2^{-2}. These values yielded sine waves with frequencies of 500 c/s and 2000 c/s respectively, with the values being generated at 10 μsec. intervals, corresponding to a sampling frequency of 100 kc/s. When the constants can be expressed as powers of 2 the computation can be carried out by shifting, rather than by the lengthy multiply or divide operation. The amplitudes of the two waveforms are set by printing in the initial values for the cosine waves at the start of the program. The initial values of the sine waves are zero. A sample of the input signal is shown in Figure 4.2

4.3 Basic Delta Modulator Program

The basic delta modulator program starts by generating the first value of the input signal as described in the previous section. From the input is subtracted the initial value of the integrated waveform e, which is usually zero. The error signal ε thus computed is used to make the decision whether to add or

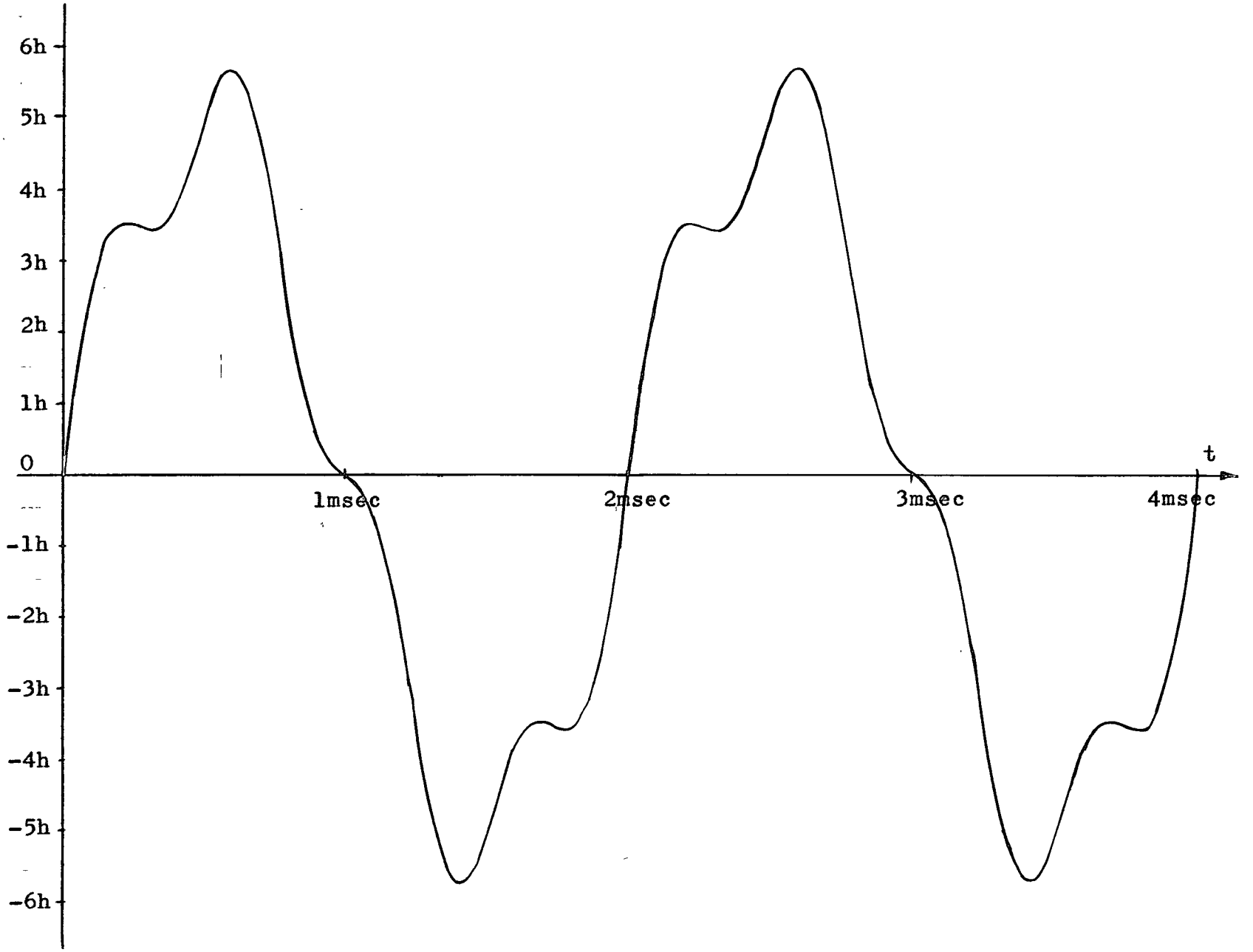


Figure 4.2 Sample Input Signal

subtract a step height h from e: if the particular value of ε is positive or zero, h is added; if negative, h is subtracted. This forms the new value of e which is ready to be subtracted from the next input. The program processes the input in exactly the same manner as an ideal physical delta modulator.

The program, along with the output routine, is reproduced in Appendix II. It is written in machine language, and the instructions are arranged in a rather complicated order. A rearranged set of commands, together with explanatory remarks, is provided in Figure 4.3 to help clarify the programming.

4.4 Remarks on the Programming

In the rearranged set of commands shown in Figure 4.3, the instructions are grouped in blocks; each of these blocks will be dealt with in turn.

The first block of instructions generates the two signals, and adds them. The next block performs the comparison, and forms ε and e. e is stored in its assigned space, but ε cannot be handled as easily. Obviously it would be impractical to store all the individual errors, since the number of storage spaces required would be equal to the number of inputs used, which in this case was usually four hundred. Instead, a "kicksorter" was programmed to perform this counting task.

4.4.1 The Kicksorter

The "kicksorter" receives the errors and sorts them according to magnitude. It records the number of errors less than one unit in magnitude, between one and two units, and so on. The units

Enter				
Clear	00	5511	2800	80
	04	4840	1704	84
Ch. II				
	08	8544	871e	88
Print in	0c	551d	5b0f	8c
$\frac{z}{h}$, $\frac{z'}{h}$	10	1160		90
	14		3000	94
	18	e740	a110	98
Store z,z'	1c	c460	178c	9c
	01	c54e	7915	81
Carriage	05	f782	8745	85
Return	09	794c	3a00	89
	0d	571e	110a	8d
Pick up y	0a	795c	1774	8a
Start	74	a505	117d	f4
loop				
	7d	615e	3132	fd
Generate	62	a505	114b	e2
y				
	4b	655c	c55e	cb
	4f	3000	7942	cf
	53	a503	111b	d3
Generate	1b	c55c	614e	9b
y'	1f	3132	11e7	9f
			a503	e7
Store e	6b	c747	c54e	eb
Pick up	6f	4140	6542	ef
h	73	3a00	3800	f3
Add d-c	77	6178	0000	f7
Add y&y'	7b	615c	1141	fb
	41	c742	11e6	c1
Form ε			6747	e6
	6a	510b	1fd0	ea
Store ε	6e	6934	2600	ee
in kick-	72	6934	11e1	f2
sorter			3600	d0
	54	683f	11e0	d4
			2600	e0
	64	683f	1161	e4
Form new e	61	3600	0c00	e1
	65	3000	6147	e5
	69	3000	674c	e9
First	71	a502	11c5	f1
Filter				

INPUT

Second			c547	c5
Filter	49	0000	614c	c9
	4d	3a00	3800	cd
	51	0000	6744	d1
	55	a502	115b	d5
	5b	c74c	1163	db
	63	6144	1168	e3
Store ε"	68	5159	1fd2	e8
in kick-	6c	6926	2600	ec
sorter	70	6926	1186	f0
			3600	d2
	56	6831	1102	d6
	02	2600	6831	82
	06	3600	3a00	86
Pick up	0a	795c	1774	8a
y				
Copy	0e	c744	8546	8e
Ch. 46 & 43	12	8143	1100	92
	00	871f	5503	80
Print	04	7840	5b07	84
out ε	08	1160	7832	88
Print	0c	5b1d	1160	8c
out ε"	10	790c	f701	90
	14	1704	5503	94
	18	0e00	7830	98
	1c	1901	1111	9c
Form	01	3000	e260	81
$\sum \bar{\varepsilon}^2$	05	c516	7841	85
	09	643f	3000	89
	0d	e654	bd16	8d
	11	1798	3000	91
Print out	15	5b06	1160	95
$\sum \bar{\varepsilon}^2$	0a	5503	0e00	8a
	0e	7832	190b	8e
Form	12	111b		
$\sum(\bar{\varepsilon}")^2$	0b	3000	e260	8b
	0f	c519	7833	8f
	13	6431	3000	93
	17	e654	bd19	97
Print out	1b	170e	3000	9b
$\sum(\bar{\varepsilon}")^2$	1f	5b1a	1160	9f
Stop	02	1102		

OUTPUT ROUTINE

Figure 4.3 Basic Delta Modulation Program

used here are decimal numbers. Twelve storage spaces were available for this purpose, so the first eleven "bins" were used to store the number of errors less than eleven units, while the last bin was reserved for the "overflow", that is, for the number of errors over eleven units in magnitude. Since the errors seldom exceeded one step height h in magnitude, setting h at eight units took care of most of the errors, neither overloading the overflow bin, nor leaving too many of the higher valued bins empty. If either of these conditions occurs, the sensitivity of the kicksorter can be changed simply by changing the value of h, increasing it to increase sensitivity (too many empty bins), or decreasing it to decrease sensitivity (overloading). The kicksorter makes up the next block of instructions.

Along with the error signal, it is desirable to have a filtered error signal, since at the receiver, the output signal is usually passed through low pass filter, and only the low frequency components of the error signal remain as the noise. To accomplish this filtering a digital equivalent of a low-pass filter was programmed, as will now be described.

4.4.2 Linear Filters

To study a linear filter for sampled data systems, analysis must be carried out in the time, rather than frequency domain. The behaviour of a linear filter can be expressed in terms of its output corresponding to a unit-step function input[5]. If, when the input is $\begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$, the output is A(t), then corresponding to any input f(t) the output is given by

$$F(t) = \int_0^{\infty} A'(\tau)f(t-\tau)d\tau + A(0)f(t)$$

where $A(0) = \lim_{t \to 0+} A(t)$.

In the discrete system this output is given by a sum

$$F(ks) = s\sum_{n=1}^{\infty} A'(ns)f(ks-ns) + A(0)f(ks)$$

where s is the sampling interval, and F(t) and f(t) are adequately determined by their values at t = ks.

In the case of a simple RC lowpass filter (Figure 4.4),

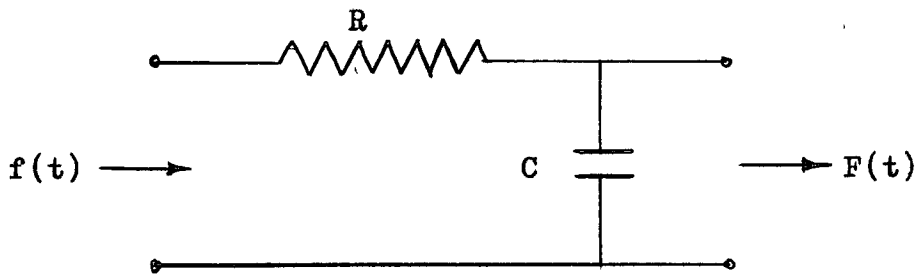


Figure 4.4 Lowpass Filter

A(t) is given by

$$A(t) = 1 - e^{-t/T}$$

where

$$T = RC$$

Also,

$$A(0) = 0$$

and

$$A'(t) = \frac{1}{T} e^{-t/T}$$

so that

$$F(ks) = s\sum_{n=1}^{\infty} \frac{1}{T} e^{-ns/T} f\left[(k-n)s\right]$$

The output can be regarded as being made up of a weighted sum of the present input and past values of input. For programming

purposes, the sum can be expressed as a recurrence relationship. In the case of the low pass filter, the past values are weighted exponentially. If the error signal ϵ is filtered to yield the filtered error signal ϵ', the recurrence relationship for ϵ' can be written down as

$$\epsilon_i' = a\epsilon_i + (1-a)\,\epsilon_{i-1}' \,, \quad a \leq 1$$

where ϵ_i is the latest value of the error signal, ϵ_i' the latest value of the filtered error signal, and ϵ_{i-1}' the latest previous value of the filtered error signal. The constant fraction a is determined by the required cut-off frequency of the filter. For this work, the cut-off frequency was set at about 4000 c/s. This gives a value for a of

$$a = \frac{2\pi(4000)}{100,000} = \frac{\pi}{12.5}$$

for a sampling frequency of 100 kc/s. For convenience in programming, $a = \frac{1}{4}$ was used, giving a cut-off frequency of 3980 c/s.

A single-stage low-pass filter attenuated the high frequencies starting at the cut-off frequency, and this attenuation increases at a rate of 6 db/octave. This is not a very sharp cut-off, and it was felt that a steeper "rolloff" was required. Hence the error signal was passed through two stages of filtering, providing a 12db/octave rolloff. The second stage of filtering is similar to the first, the signal ϵ' being used to produce the final filtered error signal ϵ''. Figure 4.5 shows a typical error signal, and the effects of the filtering.

The values of ϵ'' were stored in another kicksorter. The filters and the second kicksorter constitute the next two blocks of instructions.

The output routine completes the program. As a measure of

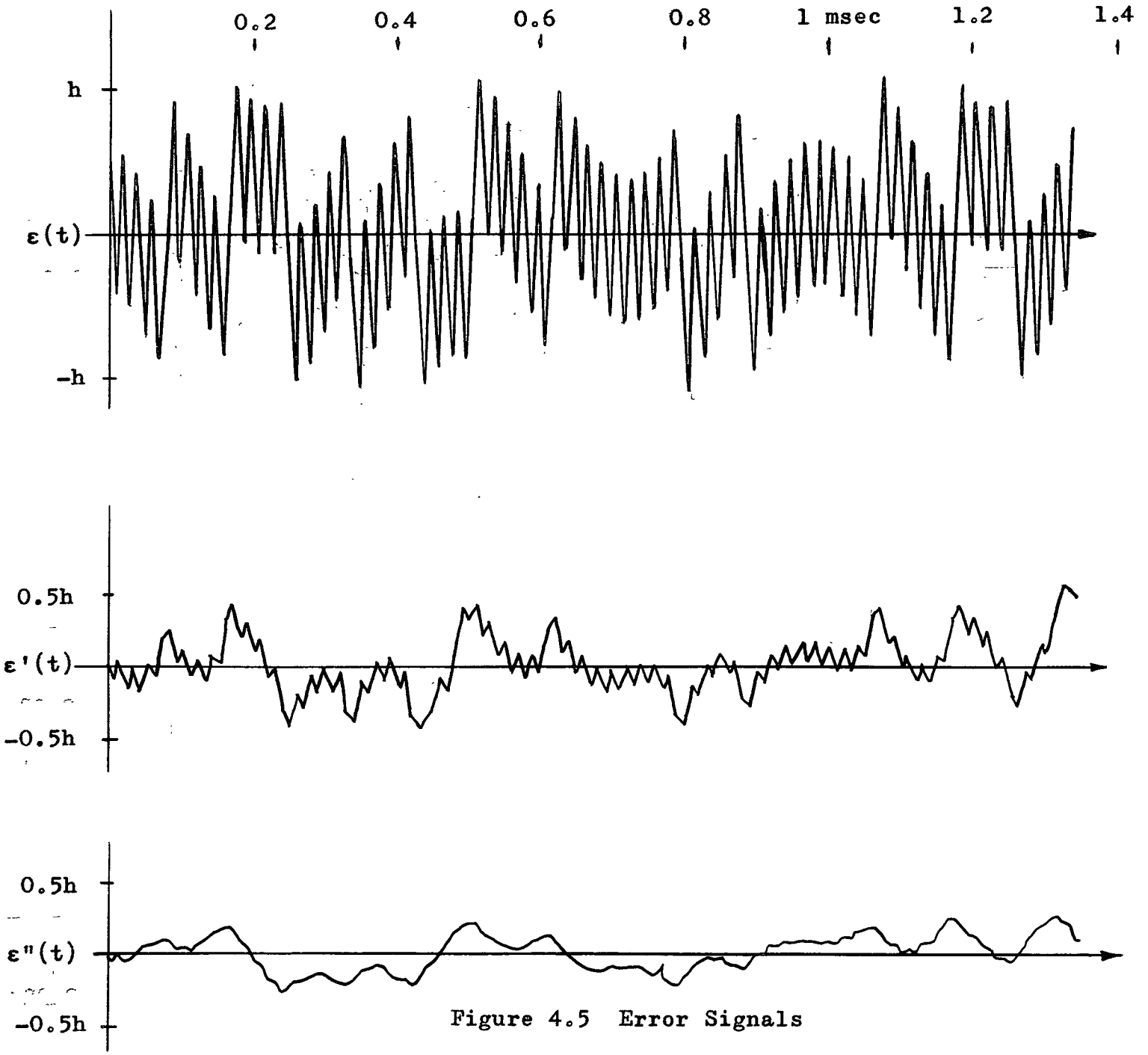


Figure 4.5 Error Signals

the average quantizing noise power, it is desirable to compute and print out the average of the squares of the errors, both for the original error signal ε and for the filtered error signal ε''. Along with noise power, it would be convenient to have some sort of a count or measure of the individual errors themselves. Hence the contents of the bins in the kicksorter were printed out. The noise power is calculated from the number of errors in each bin.

4.4.3 Calculation of Noise Power

There is no exact method of calculating the noise power of the contents of the kicksorter bins. There are several ways in which this noise power can be approximated, and one of these is described below.

The contents of the bins can be plotted as in Figure 4.6.

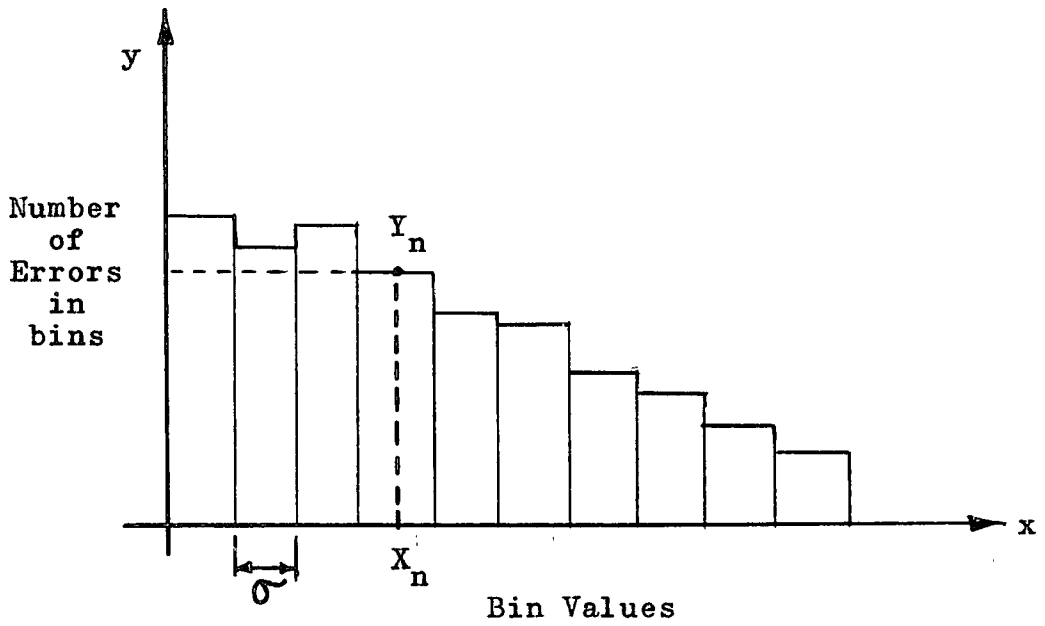


Figure 4.6 Kicksorter Contents

The simplest method would be to form the sum

$$\mu_1 = \sum_n Y_n X_n^2$$

that is, the sum of the products of the number in each bin and the average value of the bin squared. This sum, divided by the total number of errors will yield a value of the average noise power.

Another way of obtaining noise power would be to integrate the product $\frac{Y_n}{\sigma} x^2 dx$ over the interval $(X - \frac{\sigma}{2}, X + \frac{\sigma}{2})$, and sum.

$$\mu_2 = \sum_n \int_{X - \frac{\sigma}{2}}^{X + \frac{\sigma}{2}} \frac{Y_n}{\sigma} x^2 dx = \sum_n Y_n \left[X_n^2 + \frac{\sigma^2}{12} \right]$$

As before, $\frac{\mu_2}{\sum_n Y_n}$ gives the average noise power.

These methods, however, do not take into account any rapid changes in the number of errors in adjacent bins. The method finally adopted does take into account this change.

Let

$$Y_n = \frac{Y_n}{\sigma} + \frac{\delta_n}{\sigma} \left(\frac{X - X_n}{\sigma}\right)$$

where

$$\delta_n = \frac{Y_{n+1} - Y_{n-1}}{2}$$

Then

$$\mu_3 = \sum_n \int_{X_n - \frac{\sigma}{2}}^{X_n + \frac{\sigma}{2}} Y_n x^2 dx = \sum_n \int_{X_n - \frac{\sigma}{2}}^{X_n + \frac{\sigma}{2}} \left[\frac{Y_n}{\sigma} + \frac{\delta_n}{\sigma^2}(x - X_n) \right] x^2 dx$$

$$\mu_3 = \mu_2 + \sum_n \left[\frac{\delta_n}{\sigma^2} \int_{X_n - \frac{\sigma}{2}}^{X_n + \frac{\sigma}{2}} (x^3 - X_n x^2) dx \right]$$

$$\mu_3 = \sum_n \left\{ Y_n \left[X_n^2 + \frac{\sigma^2}{12} \right] + \frac{\delta_n \sigma X_n}{6} \right\}$$

The $\frac{\delta_n \sigma X_n}{6}$ values can be calculated beforehand and stored, as can the $X_n^2 + \frac{\sigma^2}{12}$ values. The program now has to take the number in each bin and multiply it by the appropriate constant, add another constant, sum, and print out this sum. This output routine constitutes the final block of instructions.

4.5 Actual Operation of Program

Several constants must be printed in before the actual program is started. The step height h must be set. In this work, h = 8 was found to be satisfactory for all the trials. The starting values of the two sine waves are printed in as part of the program. ε is set at zero.

After the program is placed in the computer, and is started, it will call for two inputs. These will be the starting values of the two cosine waves, which will determine the maximum amplitude of input into the coder. The amplitudes must be printed in through the flexowriter at the start of each run. The most useful amplitude range was found to be between 0.1h and 20h. A few runs were tried beyond this range, but did not result in much additional information. The amplitudes are printed in as multiples of h. Thus a starting input (10, 2) will produce an input to the coder with maximum amplitude about twelve step heights, and with the two frequencies mixed in an amplitude ratio of 5:1.

As it was programmed, the coder can handle four hundred input values in just over a minute. This includes the time required to process the error signal and calculate noise power, but not the printout time. At the end of each run, the computer prints out the number of errors in each bin, for both the filtered and unfil-

tered error, in two vertical columns, with the sum of the squares of the errors at the foot of each column. A sample is shown in Figure 4.7.

10 2

19.00	.00
15.00	.00
23.00	.00
30.00	.00
36.00	.00
28.00	.00
24.00	.00
37.00	.00
35.00	4.00
40.00	17.00
44.00	73.00
69.00	306.00
13876.92	376.56

Figure 4.7 Sample Computer Output

In addition to the regular program, another program was written to print out individual errors. Since this program was to be used for just one run, the time requirements were not so stringent, and the program was written using ALCOM, the automatic compiler program. This program takes a list of algebraic commands and writes a program, allotting all the necessary storage spaces automatically. It is quite inefficient as far as speed is concerned, but for this purpose, and also for printing out values of the input in one case, the compiled program worked very well. The data comes out on punched tape, and this tape was fed into a Mosely X-Y plotter. The results are seen in Figures 4.2 and 4.5.

5. VARIATIONS OF THE BASIC PROGRAM

Once the basic program was tried successfully with various input amplitudes, changes were introduced with several objectives in mind.

5.1 Adding D-C Components to the Input

The purpose of these tests was to demonstrate that the quantizing noise did not depend critically on the exact value of the input signal, as long as its amplitude was well above the threshold.

The modification was easy to carry out. After generating in the computer the values of the sine waves representing the fundamental and fourth harmonic, a constant was added to their sum. This constant was readily accessible and could be easily changed. Several values were tried.

5.2 Adding Higher Orders of Integration to the Feedback Loop

The object of these trials was to produce a system with less quantizing noise than simple delta modulation. These program changes were more difficult to carry out. As the original program had been "optimized" to save computer time, a change in the program involved re-writing a whole section of it if optimization was to be maintained.

As explained in section 3.2, systems involving higher order integration can be created by basing the decision of whether to add or subtract h from e(t), not on the polarity of the error ε, but on the polarity of a decision function d(t), which can be a combination of ε and integrals of ε. Now for the purpose of com-

puting quantizing noise, the computer is already set up to filter ε and produce ε' and ε''. (See section 4.4.2) Since this filtering process is an integration with a finite time-constant, it was decided to use ε, ε', and ε'' to form the decision function d(t).

The amended program, with provision for adding d-c to the input, and with a decision function

$$d = \varepsilon + m\varepsilon' + n\varepsilon''$$

is shown in Appendix III. Many values of m and n were tried.

5.3 Varying the Integration Time Constant

It has been assumed, both in this work and in other experiments, that the timeconstant of the integrator or integrators does not affect the amount of quantizing noise, as long as it is longer than $1/\omega_o$, where $\omega_o = 2\pi f_o$, and f_o is the cut-off frequency of the input signal. A check on this assumption was desirable.

This change required that a separate filter be programmed to operate on ε, since the output noise signal had to be processed as before, but the decision function d(t) had to be formed from a combination of ε and the signals resulting from passing ε through filters with time constants other than the ones already in use. Fortunately, there was sufficient room in the program to add two more stages of filtering, and this was done, leaving the constant a (See section 4.4.2) as a variable parameter. Several values of this parameter were tried.

6. RESULTS

6.1 Basic Delta Modulation Program

For the basic program, trials were run using maximum amplitudes ranging from 0.1h to 25h for the fundamental frequency. The results are shown in Figure 6.1. The noise is expressed as a root-mean-square amplitude and plotted versus the input signal amplitude. Both input and noise amplitudes are expressed in decibels with respect to the step-height h.

Some characteristics of simple delta modulation can be seen from the plot. The threshold effect can be seen, starting just below -5 db (maximum amplitude of 0.56h). The overload begins at about 22 db. In between these limits, the noise is irregular, but reasonably flat. In this "usable" band, the noise amplitude is on the average just below -14 db.

There are several common ways of specifying the signal-to-noise ratio of a communication system. One way is to use the maximum signal-to-noise ratio in the system; another is to use the ratio of the signal that will just overload the system to the minimum noise in the usable band; and the third method is to quote the "dynamic range" of the system, which for our purposes is taken to mean the ratio of the largest signal to the smallest signal which can be transmitted with a signal-to-noise exceeding some specified value, say 20 db.

Figure 4.2 shows the signal-to-noise ratio of the received signal for various values of signal amplitude. From this, the three signal-to-noise ratios for the system can be determined. These values were obtained:

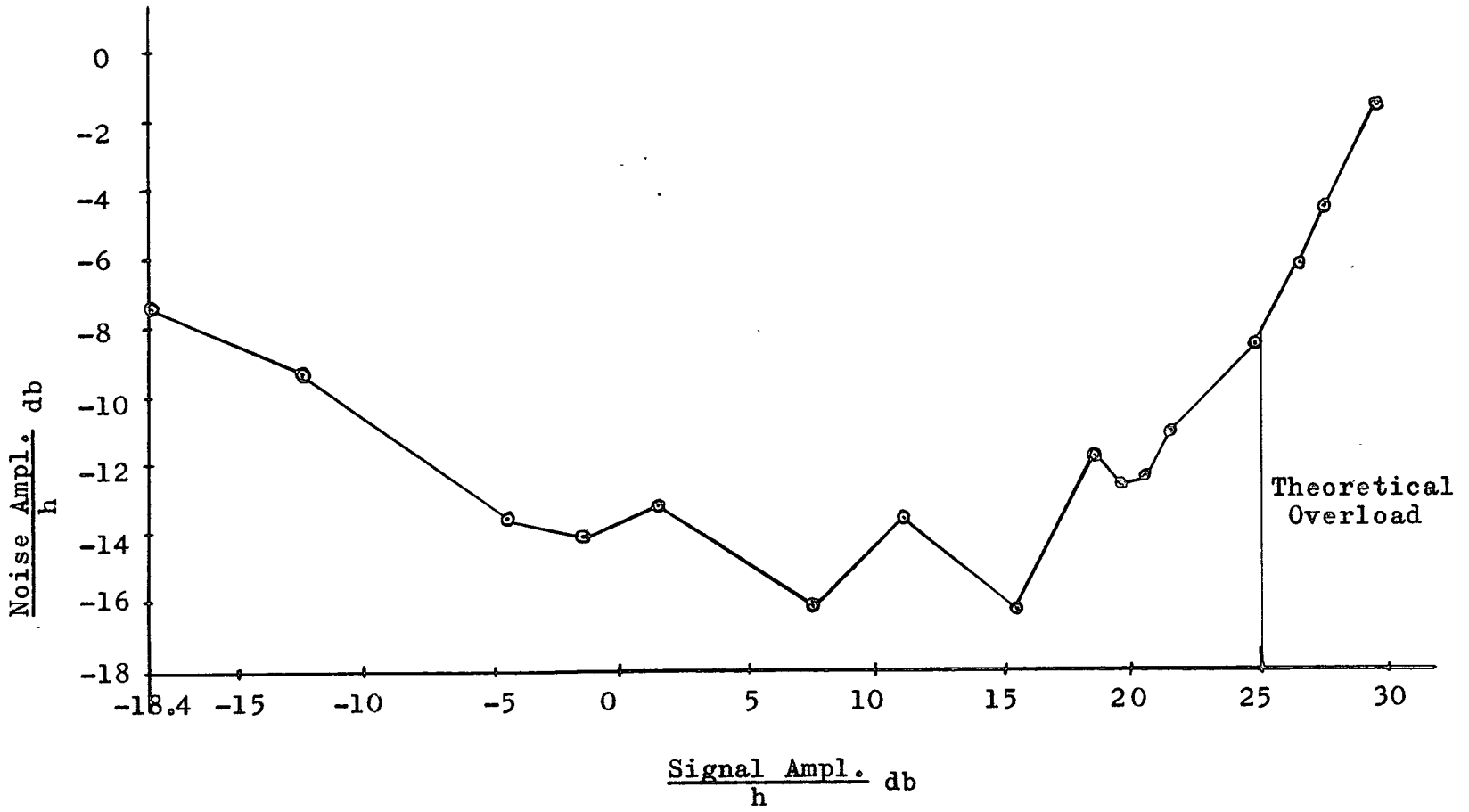


Figure 6.1 Results of Basic Delta Modulation

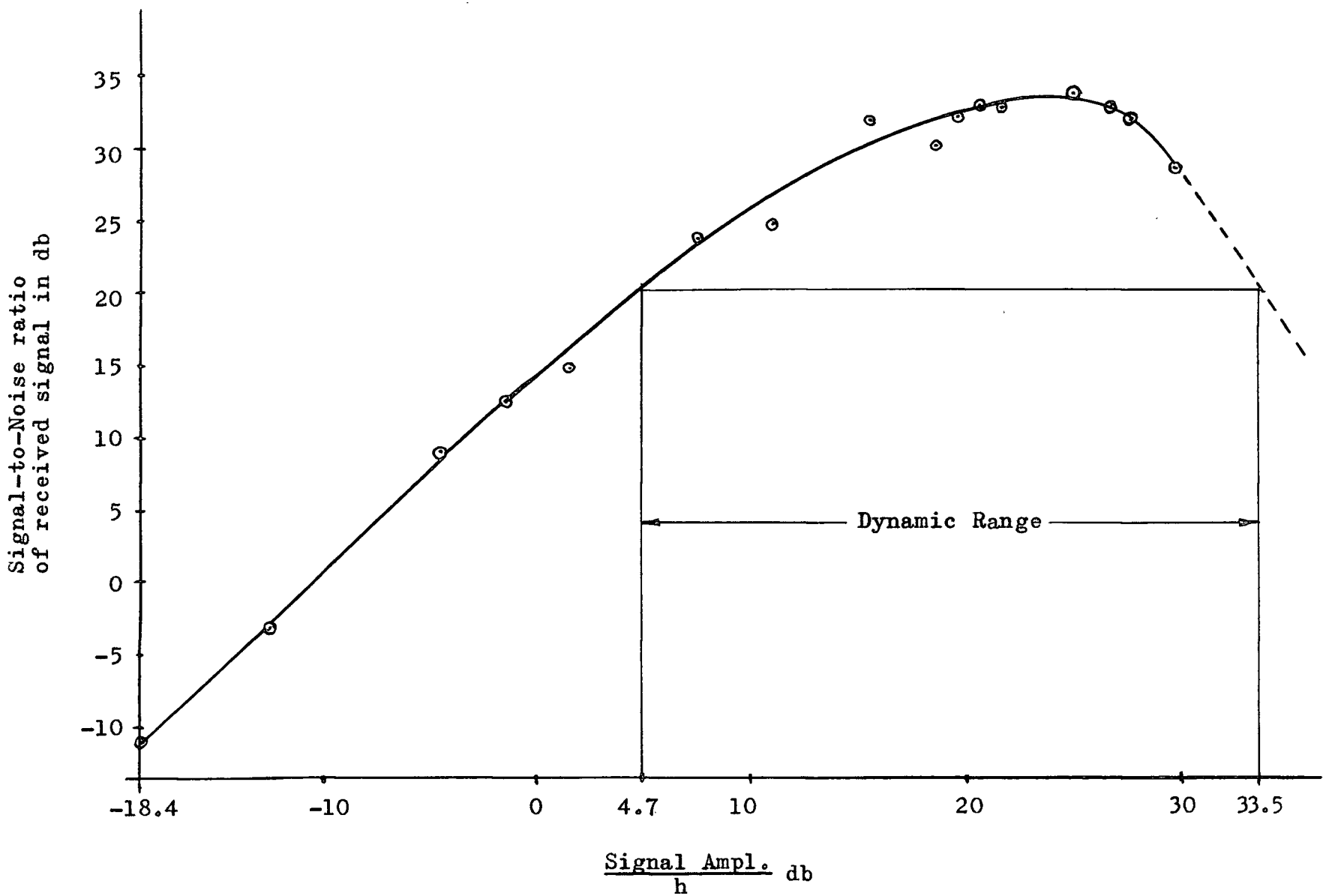


Figure 6.2 Signal-to-Noise Ratios

a) For the maximum signal-to-noise ratio in the system,

$$\left(\frac{S}{N}\right)_{max} = 33.6 \text{ db } \pm 2 \text{ db}$$

b) For the ratio of the overload signal to the minimum noise in the usable band,

$$\frac{S_{ov}}{N_{min}} = 39.8 \text{ db } \pm 2 \text{ db}$$

c) For the ratio of the largest signal to the smallest with signal-to-noise ratio greater than 20,

$$\left(\frac{S_{max}}{S_{min}}\right)_{S/N>20} = 28.8 \text{ db } \pm 2 \text{ db}$$

The value obtained in (b) can be compared with a theoretical value as calculated by de Jager[2]. He assumes a signal $A\sin 2\pi ft$, and for this signal, the overload amplitude if $\frac{f_s h}{2\pi f}$. (See Section 3.3.1). For the composite signal used in the present work, which can be written as $A\left(\sin 2\pi ft + \frac{1}{5}\sin 4(2\pi ft)\right)$, the overload occurs when the amplitude is approximately $\frac{f_s h(1+1/5)}{2\pi(1+4/5)f} = \frac{2}{3}\left(\frac{f_s h}{2\pi f}\right)$. Using this expression, the signal-to-noise ratio as derived by de Jager becomes

$$\frac{S}{N} = \left(\frac{2}{3}\right)^2 \times 0.20 \times \frac{f_s^{\frac{3}{2}}}{f \cdot f_o^{\frac{1}{2}}}$$

Substituting $f_s = 100$ kc/s, $f_o = 3980$ c/s, $f = 500$ c/s

$$\frac{S}{N} = 39.5 \text{ db}$$

This is in excellent agreement with the value found in (b) above.

6.2 Addition of D-C Component

In general, adding a d-c component to the input signal affects only the low level signal. In the usable band, there is negligible effect. The results of adding 0.1h, 0.3h, and 0.5h d-c to the input signal are shown in Figure 6.3. As might be expected, the threshold effect noise is altered appreciably, but

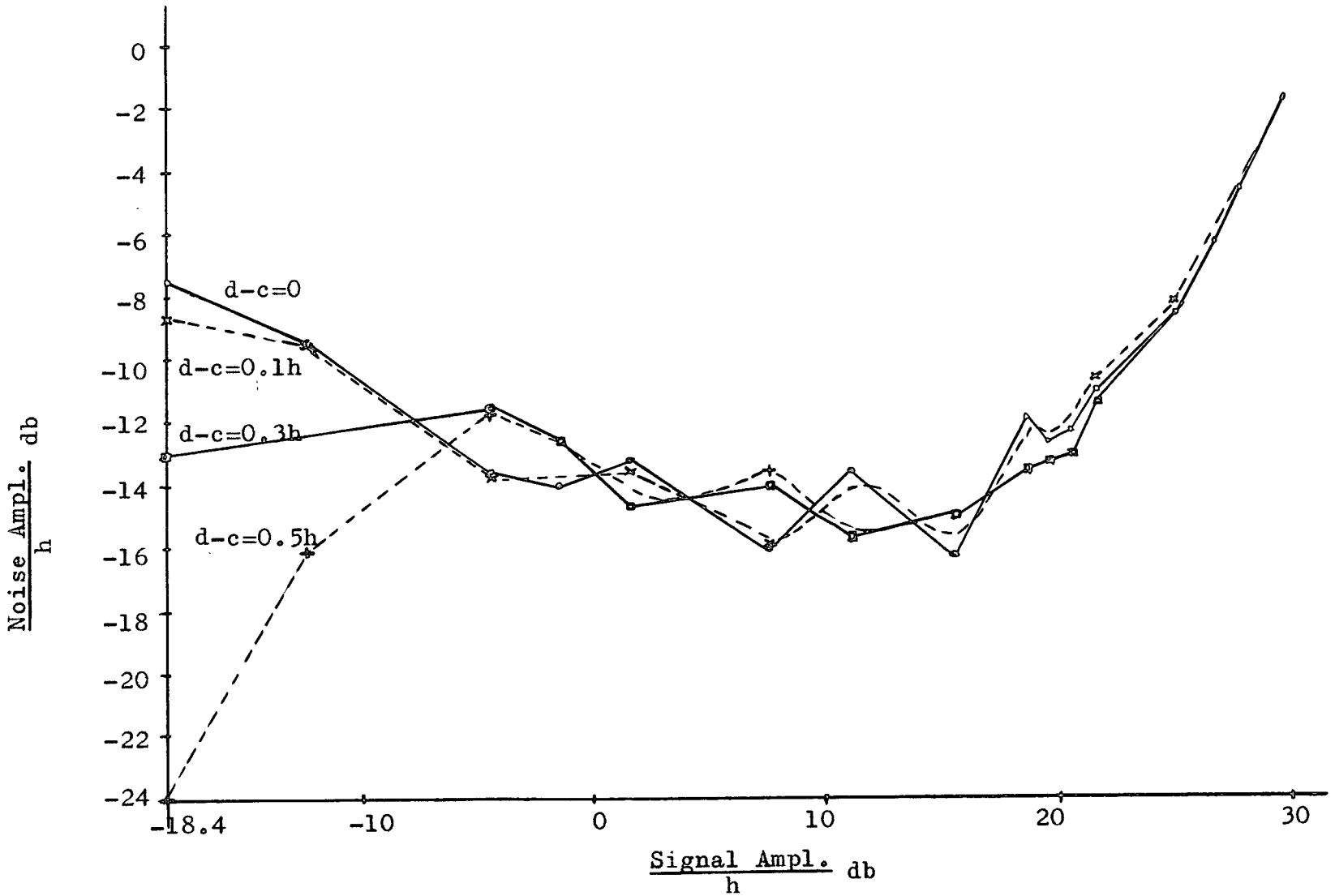


Figure 6.3 Addition of D-C Component to Input Signal

there is very little effect in the usable band, or on the overload.

6.3 Multiple Integration Systems

6.3.1 Double Integration System

At several fixed input signal amplitudes, the constant n was set at zero, and m was varied until the noise power was at a minimum. (See Section 5.2). There was a slight variation in the optimum value of the constant m at different input amplitudes, but all the optimum values of m obtained were in the range 6 to 8, and within this range, the system was quite insensitive to changes in the value of m. A value m = 7.5 was finally decided upon as being most suitable for all input signal amplitudes which would be used.

Results of trials using m = 7.5 are plotted in Figure 6.4. Here, as before, r.m.s. noise amplitude is plotted versus input signal amplitude, both in db with respect to h. Figure 6.1 is superimposed on this figure for comparison. Figure 6.5 shows signal-to-noise ratio plotted versus signal amplitude.

As can be seen from Figures 6.4 and 6.5, this optimized double integration system has an improvement of noise level in the usable band of about 7 db. The threshold effect is virtually eliminated, extending the dynamic range of the system by about 7 db.

This result can be compared to the optimum double integration system cited by de Jager[2]. The first point of comparison is between the optimum m and his optimum τ. (See Section 2.2). Now τ can be expressed in terms of m as

$$\tau = \frac{T_2}{m},$$

where T_2 is the time constant of the filter which integrates e.

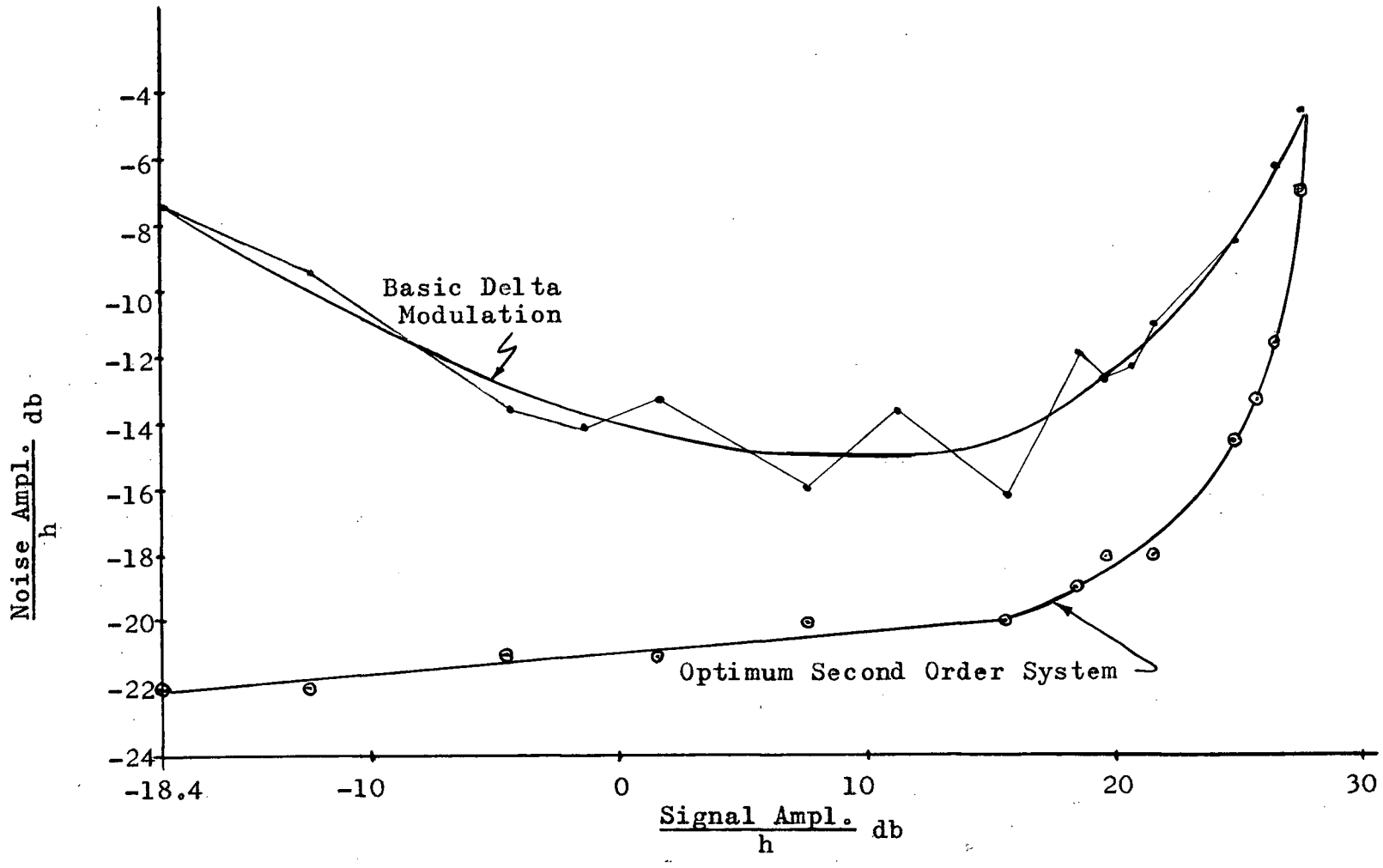


Figure 6.4 Optimum System

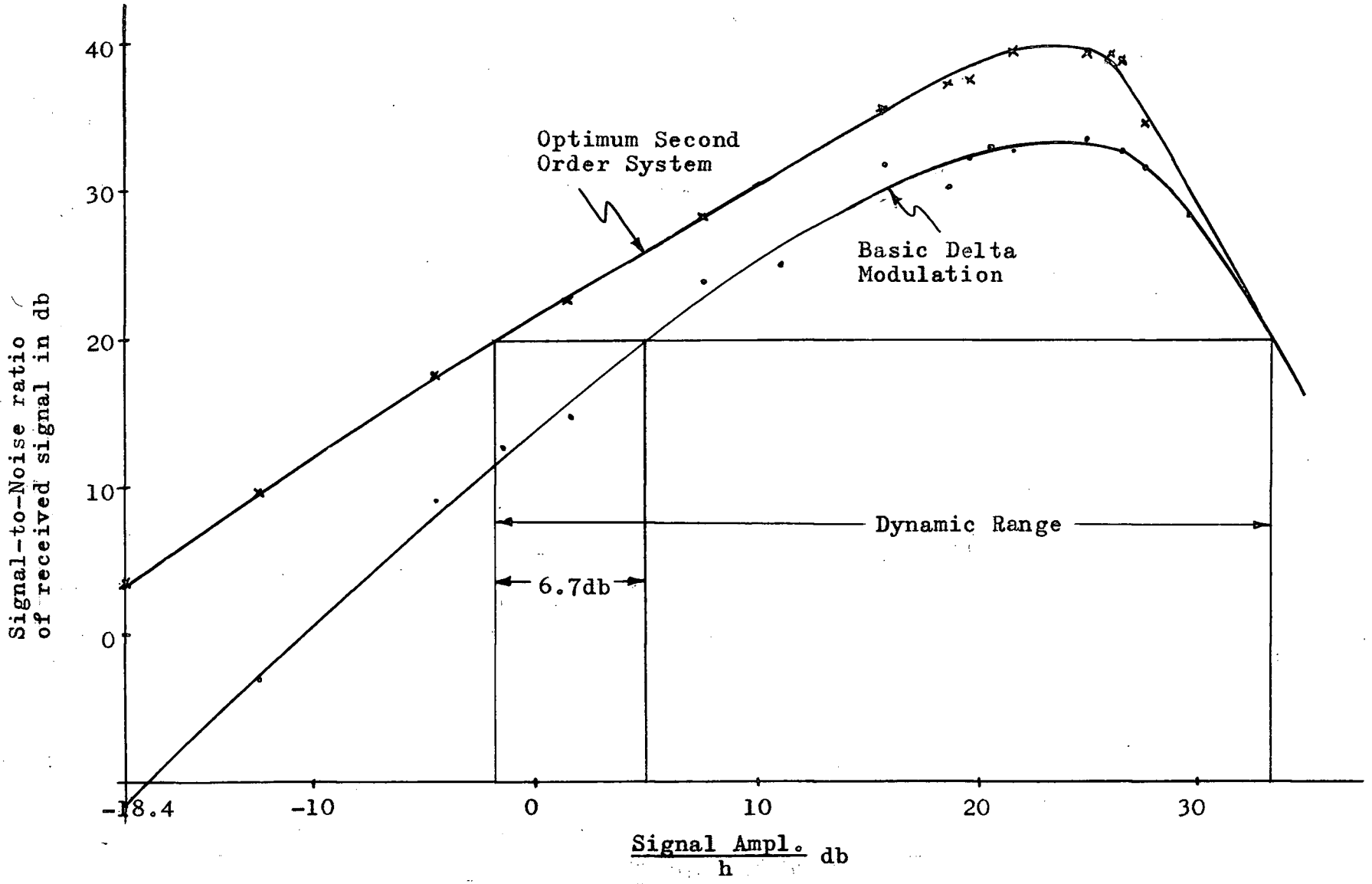


Figure 6.5 Signal-to-Noise Ratios for Optimum System

For this filter,

$$T_2 = \frac{1}{3980(2\pi)}$$

so the optimum T is given by

$$\tau_{opt.} = \frac{T_2}{m_{opt.}} = 5.35\ \mu\ \text{sec.},$$

using $m_{opt.} = 7.5$. Now according to de Jager, the optimum τ is about one-half the sampling time, or in this case, 5 μ sec. The value found in these trials is in very good agreement with de Jager's value.

The second point is the improvement in the system due to introducing second order integration. De Jager quotes a figure of 10 db as his improvement, and this figure is slightly higher than the improvement shown in Figure 6.5.

6.3.2 Adding Third Order Integration

The next step was to form the decision function d(t) using portions of first and second integrations of ϵ. (See section 5.2). Accordingly, for fixed values of input signal amplitudes, the constants m and n were varied. At most lower values of input signal, there was little or no improvement due to adding the second integration of ϵ. Slight improvement at higher input amplitudes were obtained, but at the expense of higher noise amplitudes at the middle valued inputs. The best results were obtained with m = 5 and n = 1. These results are shown in Figure 6.6, together with the optimum second order system for comparison. It was felt that these results did not merit further investigation of this line of attack.

6.4 Changing the Filter Time Constant

In the original filters used to filter ϵ, the constant a

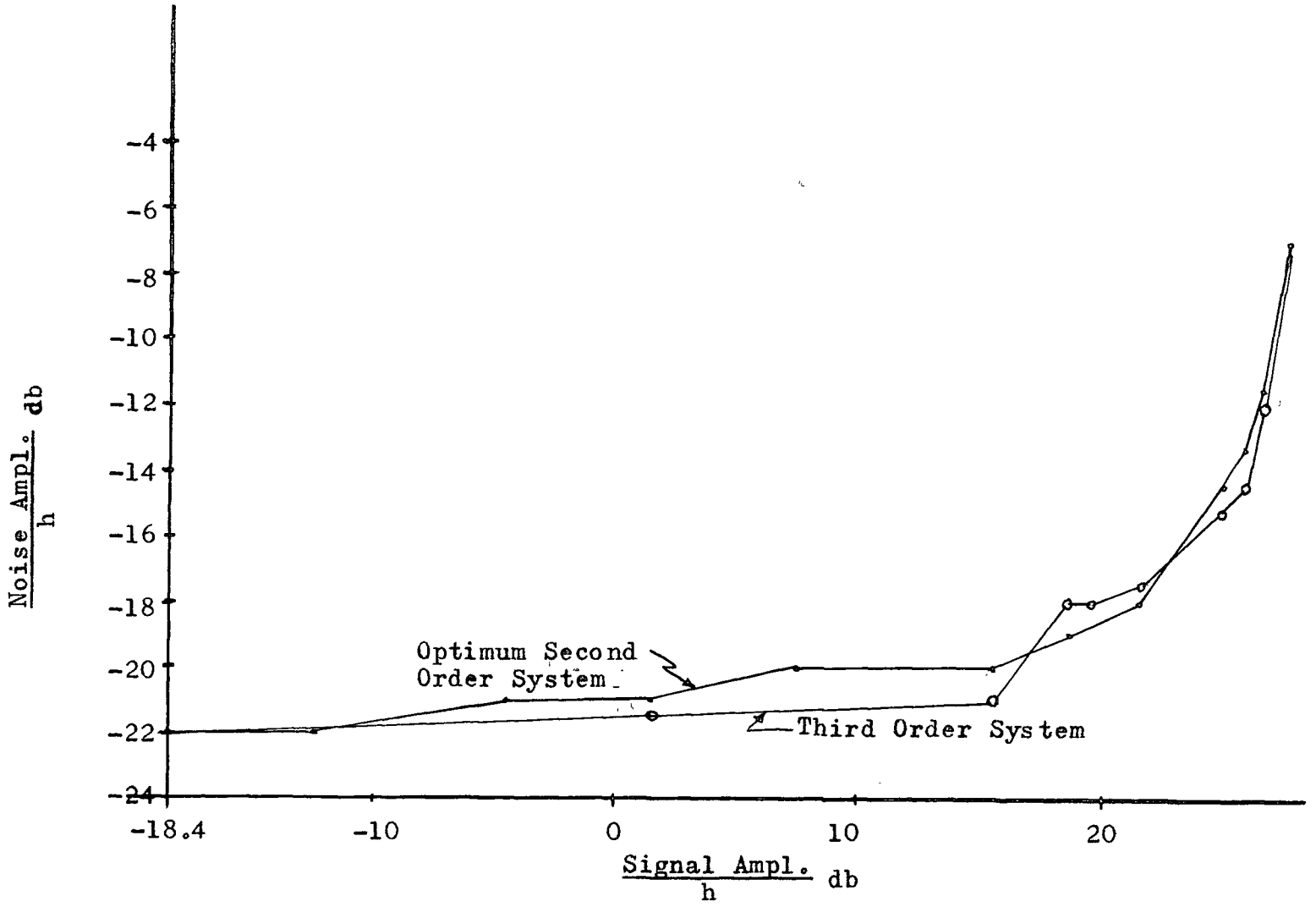


Figure 6.6 Adding Third Order Integration

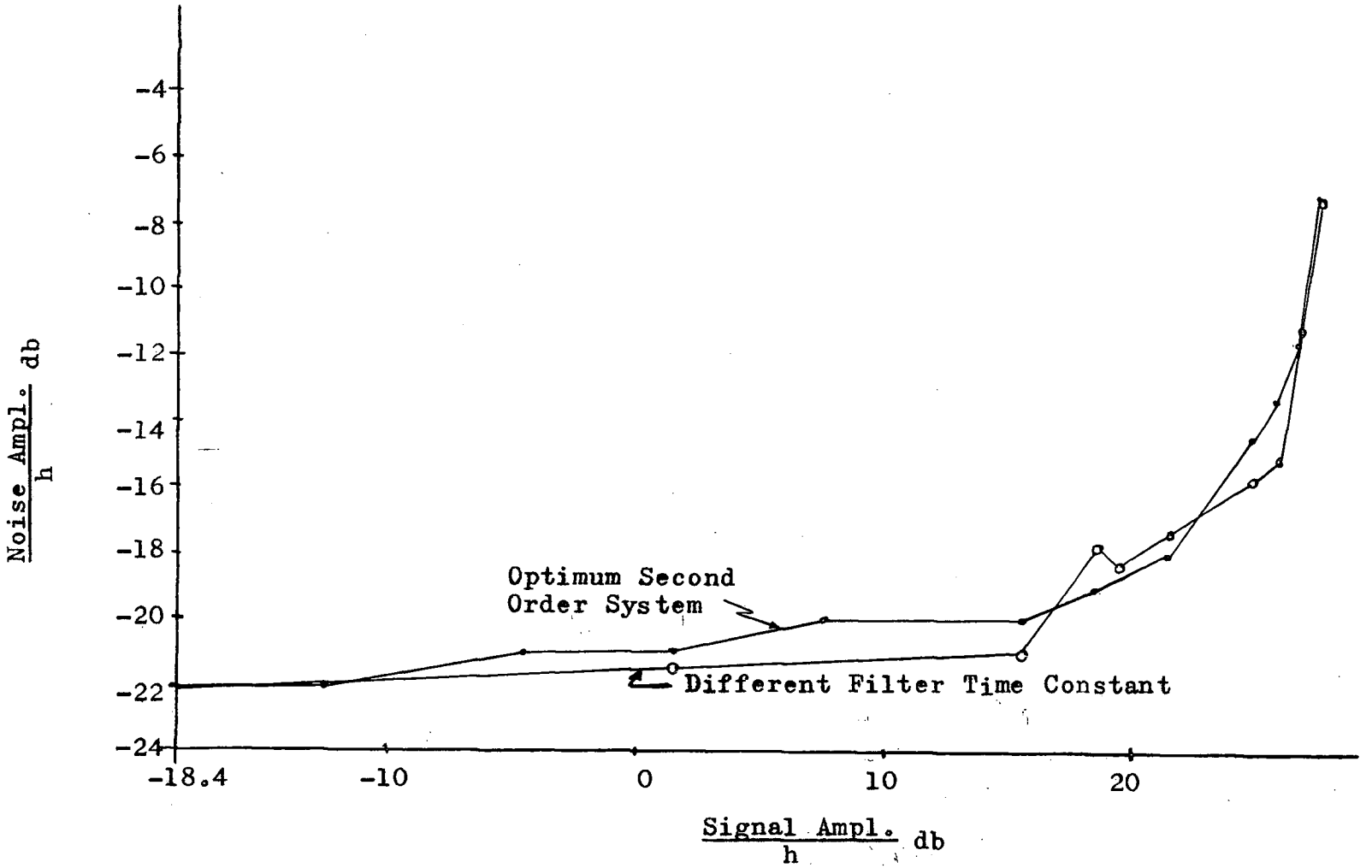


Figure 6.7 Changing the Filter Time Constant

was set at $\frac{1}{4}$, giving a cut-off frequency of 3980 c/s. (See section 4.4.2). For these trials, the values $a = \frac{1}{8}$ and $a = \frac{1}{16}$ were tried in the filters which generated the integrated errors for the decision function. These values represent cut-off frequencies of 1990 c/s and 995 c/s respectively. For each value, a new optimum m had to be found.

The first change, with m = 5, produced no significant change. There was slight improvement for some amplitudes, and a slight worsening of the noise for others. These results are illustrated in Figure 6.7. The second change seemed to increase the noise power at all values of input, and so these trials were abandoned.

7. CONCLUSIONS

7.1 Conclusions Regarding the Delta Modulation System

The computed behaviour of simple delta modulation agrees well with that previously observed, for all input levels from threshold to overload. An appreciable improvement in quantizing noise, (both at normal and at very low signal levels) results from the addition of a proportion of double integration to the feedback loop. The optimum proportion has been determined and is recommended for all delta modulation systems used for speech signals.

The addition of some third-order integration gives an improvement which is barely detectable, and probably not worth while. The time-constants of the integrators are not critical, and may conveniently be as short as 40 μsecond.

7.2 Conclusions Concerning Digital Simulation Techniques

Simulation of a digital communication system on a digital computer can be done with great accuracy, with freedom to alter parameters of the system easily, and with no fear that results are influenced by imperfections in the apparatus. Furthermore criteria for transmission quality (such as the quantizing noise power) can be computed as part of the program with greater accuracy and reliability than can be expected of experimental measurements.

For a computer such as the ALWAC III-E, with modest speed and storage facilities, the small sample of the input signal to be processed must be carefully chosen, and the program painfully optimized for reasonable computing times. Results of such trials are more questionable, but can still give much valuable

information in a short time. The work described here shows that where comparison with experiment is possible, the results are quite accurate. Such computer simulation trials are recommended as a method of selecting the best of several possible communication systems, though the final choice should be verified with a "listening test".

APPENDIX I

GENERATING A SINUSOIDAL FUNCTION

The method used to generate values of a sine wave at intervals of Δt is the simultaneous solution of a pair of differential equations. Starting with

$$y = A \cos\omega t \quad \text{and}$$

$$z = A \sin\omega t, \qquad \ldots\ldots \quad (1)$$

we differentiate to get

$$\dot{y} = -\omega A \sin \omega t = -\omega z$$

$$\text{and} \quad \dot{z} = \omega A \cos \omega t = \omega y \qquad \ldots\ldots \quad (2)$$

To generate values at intervals of Δt, we can use

$$\Delta y = -kz$$

$$\text{and} \quad \Delta z = ky \quad \text{where } k = \omega \Delta t \qquad \ldots\ldots \quad (3)$$

For the iteration process, there are several possibilities:

a) $y_{n+1} - y_n = -kz_n$

$z_{n+1} - z_n = ky_n$

b) $y_{n+1} - y_n = -kz_{n+1}$

$z_{n+1} - z_n = ky_{n+1}$

c) $y_{n+1} - y_n = -kz_n$

$z_{n+1} - z_n = ky_{n+1}$

In general, $y_{n+1} = y_n - kz_n - \ell y_n$

$$z_{n+1} = z_n - ky_n - mz_n \qquad \ldots\ldots \quad (4)$$

where l and m are constants with values either 0 or k^2.

Assume y_n and z_n to of the form

$$y_n = B\lambda^n \quad \text{and}$$

$$z_n = C\lambda^n \quad \text{where } \lambda \text{ is a complex number} \quad \ldots \quad (5)$$

Then from (4) and (5)

$$B\lambda^{n+1} = B\lambda^n - kC\lambda^n - \ell B\lambda^n \,, \quad \text{and}$$
$$C\lambda^{n+1} = C\lambda^n + kB\lambda^n - mC\lambda^n$$

or

$$\lambda = 1 - k(\tfrac{C}{B}) - \ell \,, \quad \text{and}$$
$$\lambda = 1 + k(\tfrac{B}{C}) - m$$

So that

$$\frac{C}{B} = \frac{k}{\lambda + m - 1}$$

and

$$\lambda = 1 - \frac{k^2}{+ m - 1} - \ell$$

Bring all the terms to the left-hand side, we have

$$\lambda^2 - \lambda(2-\ell-m) + 1 + k^2 + \ell m - \ell - m = 0 \qquad \ldots\ldots (6)$$

The interesting case is when $\ell = 0$, and $m = k^2$(Case c)

then

$$\lambda^2 - \lambda(2 - k^2) + 1 = 0$$
$$\lambda = 1 - \frac{k^2}{2} \pm ik\sqrt{1 - \frac{k^2}{4}} \qquad \ldots\ldots\ldots (7)$$
$$|\lambda| = \sqrt{1 + \frac{k^4}{4} - k^2 + k^2 - \frac{k^4}{4}} = 1$$

So that one solution is $\lambda = e^{i\theta}$ $\qquad \ldots\ldots\ldots (8)$

Then $Y_n = e^{in\theta}$, letting $A = 1$

$$Y_n = (\cos\theta + i\sin\theta)^n = \cos n\theta + i\sin n\theta \qquad \ldots\ldots\ldots (9)$$

So that using case (c), a pure stable sinusoidal signal may be generated.

The relationship between $n\theta$ and ωt may be investigated. Equating the real and imaginary parts of (7) and (8) we have

$$\cos\theta = 1 - \frac{k^2}{2}$$

and

$$\sin\theta = k\sqrt{1 - \frac{k^2}{4}} \qquad \ldots\ldots\ldots (10)$$

$$1 - \frac{k^2}{2} = \cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \ldots\ldots$$

$$k^2 = \theta^2\left(1 - \frac{\theta^2}{12} + \ldots\ldots\right)$$

$$k = \theta\left(1 - \frac{\theta^2}{12} + \ldots\ldots\right)^{\frac{1}{2}} = \theta\left(1 - \frac{\theta^2}{24} + \ldots\ldots\right)$$

$$\theta = \frac{k}{1 - \frac{\theta^2}{24} + \ldots\ldots} = k\left(1 + \frac{\theta^2}{24} - \ldots\ldots\right)$$

$$\theta = k\left\{1 + \frac{\left[k\left(1 + \frac{\theta^2}{24} - \ldots\ldots\right)\right]^2}{24} - \ldots\ldots\right\}$$

$$\theta = k\left(1 + \frac{k^2}{24} + \ldots\ldots\right)$$

$$n\theta = nk\left(1 + \frac{k^2}{24} + \ldots\ldots\right) = n\omega\Delta t\left(1 + \frac{k^2}{24} + \ldots\ldots\right)$$

$$= \omega t\left(1 + \frac{k^2}{24} + \ldots\ldots\right) \qquad \ldots\ldots\ldots \quad (11)$$

APPENDIX II

BASIC DELTA MODULATION PROGRAM

The program is stored in channels 42 to 46 inclusive.

```
4204
55112800 c54e7915 26006831 00000000
48401704 f7828745 36003a00 00000000
8544871e 794c3a00 795c1774 000b0000
551d5b0f 571e110a c7448546 00100094
11600000 00200000 81431100 00000000
00003000 5b680000 00000000 00000000
e740a110 81431100 00000000 c55c614e
c460178c 00020000 01910000 313211e7
4304
871f5503 3000e260 11020000 000c0000
78405b07 c5167841 0608020a 06000288
11607832 643f3000 55030e00 3000e260
5b1d1160 e654bd16 7832190b c5197833
790cf701 17983000 111b0000 64313000
17045503 5b061160 00000000 e654bd19
0e007840 00000000 06080202 170e3000
19011111 06000210 00000000 5b1a1160
4404
00080000 c74211e6 00000000 00000000
00000000 0000c547 00000000 00000000
00000000 0000614c 00000000 655cc55e
00000000 3a003800 00000000 30007942
00003600 00006744 00003600 a503111b
683f11e0 a502115b 68311102 00000000
00000000 000b0000 00000000 c74c1163
00000000 00000000 00000000 00000000
4504
00002600 36000c00 a505114b 61441168
683f1161 30006147 00006747 0000a503
51591fd2 3000674c 510b1fd0 c744c54e
69262600 a50211c5 69342600 41406542
69261186 00000000 693411e1 3a003800
a505117d 00000000 00000000 61780000
00000000 00000000 00000000 615c1141
00000000 615e3132 000b0000 00000000
4604
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
000000f5 000000e0 000000cb 000000b5
000000a0 0000008b 00000075 00000060
0000004b 00000035 00000020 0000000b
00008455 00006e55 00005a55 00004855
00003855 00002a55 00001e55 00001455
00000c55 00000655 00000255 00000055
```

APPENDIX III

DELTA MODULATION PROGRAM

with $d(t) = \varepsilon + m\varepsilon' + n\varepsilon''$

```
42
83470000 c55a7913 00000000 30001775
00000000 f7828745 00000000 c74e8546
8544871e 28000000 0000615d 81431100
551d5b0f 571e111b 0c003000 00100094
11600000 00000000 111b0000 5b680000
00003000 00000000 00000000 00000000
e746a110 00000000 00000000 415c614e
c460178c 00020000 01910000 3a001103
43
871f5503 3000e260 11020000 000c0000
78405b07 c5167841 0608020a 06000288
11607832 643f3000 55030e00 3000e260
5b1d1160 e654bd16 7832190b c5197833
790cf701 17983000 111b0000 64313000
17045503 5b061160 00000000 e654bd19
0e007840 00000000 06080202 170e3000
19011111 06000210 00000000 5b1a1160
44
000c0000 00000000 00000000 00000000
36006831 3a003800 00080000 00000000
26001152 614a615c 00000000 00000000
00000000 674e5140 00000000 51401f44
00000000 c7421f66 68313600 69262600
00000000 69342600 495b117e 69261156
00000000 69341lee 00000000 00000000
00000000 00000000 00000000 00000000
45
00000000 00000040 a505116b c5244165
c7650000 00000000 3600683f e7791121
615b114f 615ac55c 2600683f 655cc55e
c75d1174 30003200 36003a00 30007942
00000000 a50311f7 38006765 a5031169
61653a00 a505117d a502116c 0000c55a
3800675b 00000040 00000000 0000c74e
a5021164 615e3132 3000e761 65421145
46
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
000000f5 000000e0 000000cb 000000b5
000000a0 0000008b 00000075 00000060
0000004b 00000035 00000020 0000000b
00008455 00006e55 00005a55 00004855
00003855 00002a55 00001e55 00001455
00000c55 00000655 00000255 00000055
```

47

```
00000000 30006124 00000000 00000000
00000000 4146118a 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
```

REFERENCES

1. De Faye, P.J., "Aspects of Delta Modulation," M.A. Sc. Thesis, University of British Columbia, 1959.

2. De Jager, F., "Deltamodulation, A Method of P. C. M. Transmission Using the 1-Unit Code," Phillips Research Reports, Vol. 7, pp. 442-466, 1952.

3. Holzer, J., "Exponential Delta Modulation for Military Communications," Signal Corps Engineering Laboratories, Fort Monmouth, N.J., Technical Memorandum No. M-1777, 1 June 1956.

4. David, E.E. Jr., Mathews, M.V. and McDonald, H.S., "Experiments With Speech Using Digital Computer Simulation," Bell Telephone System Technical Publications, Monograph 3405.

5. Levinson, N., "The Wiener RMS Error Criterion in Filter Design and Prediction," Journal of Math and Physics, Vol. XXV, No. 4, pp. 261-278, January 1947.