DATA REDUCTION TECHNIQUES FOR HIGH-QUALITY
DIGITIZED AUDIO

preprint no. 1443 (G-5)

by

JAMES A. MOORER, IRCAM, Paris, France

Presented at
the 62nd Convention
March 13 through 16, 1979
Brussels, Belgium

This preprint has been reproduced from the author's advance manuscript, without editing, corrections or consideration by the Review Board. For this reason, there may be changes should this paper be published in the Journal of the Audio Engineering Society. Additional preprints may be obtained by sending request and remittance to the office of Special Publications, Audio Engineering Society, 60 East 42nd Street, New York, New York 10017, USA.

AN AUDIO ENGINEERING SOCIETY PREPRINT

INTRODUCTION

With the ever-decreasing prices of digital hardware and storage, it would appear that a wave of interest in the use of digital media for storage of high-quality audio is mounting. The current round of digital tape recorders is but one example of this interest [Myers and Feinberg 1972, McCracken 1978]. Synthesis of music via computer is now a well-known practice [Moorer 1977, Mathews 1969] which is established largely in universities around the world. Digital synthesis hardware for musical purposes is now coming to fruition as we see several different designs for synthesizers developing [Alles and diGiugno 1977, Alonso et al 1976]. There is even digital hardware available for reverberation simulation on the commercial market [Blesser et al 1975]. It requires at this time only a small leap of faith to envision digital processing all the way from microphone preamplifier to home loudspeaker. This being the case, it seems reasonable to ask whether the currently popular method of storing this data, that is, in PCM, is necessarily the best. We clearly need good guidelines, based on perceptual studies, as to what defines quality in digitally coded sound.

Almost all commercial digital audio is stored in terms of PCM. The sampling rates vary (on the commercial, high-quality units) from a low of about 32 KHz [Alles and diGiugno 1977, Warnock 1976, Blesser et al 1975] to a high of 50 KHz [McCracken 1978]. The number of bits in the code word is almost universally 16, although this is sometimes realized by converting less than that, such as 12 bits, and using scale switching to achieve the dynamic range [Blesser et al 1975, Kriz 1975]. There is some evidence that a higher dynamic range than the potential 96dB range of the 16-bit word might be nice, but certainly not economical at this time.

We can debate endlessly about the choice of different numbers of sampling rates and different numbers of bits. Users of the lower sampling rates offer that most people cannot hear above a certain frequency, and that the phase linearity of digital audio makes the lack of high frequencies much less audible. Another argument for lower overall quality is the one often given that most music is presented finally in home stereo systems which have somewhat limited fidelity in themselves, or that the public at large does not even appreciate higher fidelity. There is always the example of automobile music systems and portable cassette players to fix the minimum quality currently in use today. Other arguments by users of lower sampling rates are that the hardware to process the higher sampling rates is prohibitively expensive, or in the case of digital audio tape, the data density on the tape (typically 28,000 bits per inch) even at lower sampling rates is already at the limits of the technology. Any argument based on hardware costs or limitations will evaporate with time since the prices of digital hardware show no signs of ceasing their exponential downward trend in the years to come. The problem of high data densities is more fundamental, in that higher densities mean greater error rates.

In our own discussions with the people who record, make, and produce music today, it is clear that they will not abandon the current technology for the newer digital hardware until they can achieve with the same ease all the things they can do with contemporary professional audio equipment. This is quite a tall order for digital competition. For example, if we do not count the microphone preamplifiers, the specifications for modern mixing consoles exceeds a 110 dB total dynamic range. This would imply an 18 to 19 bit system which is clearly beyond the bounds of modern conversion systems today. If we just consider the tape equipment, with noise reduction devices a 90 dB signal to noise ratio can often be achieved, although it is impossible to get rid of the few tenths of a percent of 3rd harmonic distortion that is inherent in magnetic analog recording. As surprising as it may seem to people thinking about making digital audio equipment, there are recording engineers, in the popular music field, who are capable of distinguishing a recording from an identical recording with all the high frequencies above 15 KHz removed. Thus there is reason to believe that whatever we may think about its advantages, digital audio will not find a large place in the professional audio market until it can compete in quality and convenience, although not necessarily in price, with current professional audio systems.

Despite the raging interest in digital audio, there seems to be very little published material on the perceptual bases for coding schemes for high-quality audio [Lee and Lipschutz 1976]. It is not really well known how much distortion the ear can tolerate or what kind of audio material should be used to test digital audio systems. It is the purpose of this article to make a micro-step toward not necessarily the answers to these questions, but toward an understanding of what are the issues are and what are the right questions to ask.

Since our own investigation of this question stems from the investigation of specific coding schemes for high-quality audio, we shall begin our discussion with these coding schemes.

MINIMUM-LENGTH CODING

The fundamental idea of Huffman coding is that the numbers which represent (in our case) the sound samples are not evenly distributed: certain numbers occur more frequently than others. This being the case, we can assign very short codes to the numbers that occur most often, and longer codes to the numbers that occur less often. This is usually accomplished in two passes over the sound: on the first pass, a histogram is computed giving the frequency of occurance of each number. From this histogram, we can compute the code using the method described in Huffman [1952]. On the second pass, we substitute the minimum-length code for each sample. This has the feature that the samples no longer take the same amount of storage. The first sample might be 2 bits long, the next sample 6 bits, and so on. The advantage of this system is that there is no loss of information. The coding scheme can encode any string of numbers without error, thus no questions of perceptibility of coding distortion can arise.

There are several disadvantages, however, to this coding scheme. One of the most fundamental is that the optimum coding can not be determined until one has the entire signal available to compute the histogram. The severity of this limitation, though, depends on the application. For recordings which have been made in the conventional way but are now to be distributed in digital form, this presents no problem in that the code can be computed before the distribution is done, after the recording is completed. This does mean that the code table must be transmitted as a preamble to the sound samples themselves. For doing direct digital recording, the statistics of the signals are not available beforehand, and some pre-computed table must be used. This guarantees a sub-optimal coding, and in some cases can actually explode the data rate above that required for straight PCM coding.

To further explore the utility of sub-optimal coding, we embarked on an empirical study. We coded several different digitized samples of speech, computer-generated music, and live music in various different ways to try to compare the results. Furthermore, there was some suspicion that coding first or second differences might increase the redundancy in the samples because of the fact that ordinary music tends to have most of its energy in the lower frequencies, thus implying smaller differences. This is not necessarily true, however, of computer-generated sound. The composer of computer music can use full-amplitude signals of any frequency.

The coding scheme we chose is, in fact, a modification of the Huffman scheme, in that for every code that takes more than the original number of bits, we substitute a special code followed by the original sample. In the worst cast, by complete mismatch of code and sound, the data would be increased by a factor of two, but no more. In Table I, we show the results for three different sound files with varying number of high-order bits. Our first observations were that the histograms for positive numbers were virtually identical to those for negative numbers, so that no additional reduction was obtained by treating them differently. Thus to code the sample, we first store the sign bit then take the absolute value. Our next observation was that the entropy of the low-order bits was so high that it only made sense to code the high-order bits.

In table I, the sounds used were LS, an utterance by a male speaker in a dry environment, PD, a piece of computer-processed music with live flute, synthetic and natural voice, other computer effects, and simulated reverberation, and TR, a piece of computer-synthesized speech music with reverb. The rows marked 0, 1, and 2 refer to 0th difference (no change), 1st differences, and 2nd difference, which is expressed mathematically as follows:

$X(n)$	0th difference
$X(n)-X(n-1)$	1st difference
$X(n)-2X(n-1)+X(n-2)$	2nd difference

These are equivalent to pre-emphasis filters and require the appropriate de-emphasis filter for decoding. The exact inverse filters for these are quite simple to construct, requiring only memory, additions, and shift operations with no multiplications.

The first question we ask is how many high-order bits does it make sense to code? If we think in terms of a 16-bit 2's complement input sample, we can see that in most cases, the mean sample length increses exactly in correspondance with the number of high-order bits used. A slight economy is realized in going from 10 bits in sound file TR to 12 bits, but virtually no economy is realized in going from 12 to 13 or to 14. This implies that 12 bits is a reasonable number of high-order bits to use in the coding. The sign and the low-order 3 bits should be transmitted separately in an uncoded format.

The next question is what order difference should be used? This seems to depend strongly on the sound. PD, which is rich in high-frequencies, shows the least improvement in going from 1st to 2nd differences, but reports nonetheless substantial improvement in going from 0th to 1st differences. Since the other files show improvement in going to 2nd differences, it seems worthwhile to use 2nd differences universally.

The final question we will address here is that of the loss in using sub-optimal codes. To test this, we used four more sound files and computed all combinations of codings and histograms. Sound file DU was computer synthesized bell-like tones with no reverberation. SC was music instrument tones that had been analysed and synthesized using additive synthesis, then reverberated by the computer. FX is a digital recording of a live flute in a very dry environment. FW is this same flute reverberated by direct convolution with a synthetic concert-hall impulse response of quite long duration. FW is, in fact, very typical of popular classical recordings. There is often a great deal of reverberation present in later classical works.

The rows of table II are codings of different sound files with a table that has been computed as optimal for the sound file whose name is at the left. The columns are the different codings of a given sound file. The diagonal represents the optimal coding of a sound file, that is, by its own code. We can see clearly that the mismatch of code to sound file is, in some cases, disasterous. Coding of FW by any but its own code produces an explosion of the amount of data involved. The reduction by use of its own code is, in fact, minimal and hardly worth the effort. The other sound files seem to reduce satisfactorily, however, and if one can tolerate the occaisional data expansion, then this may be indeed a useful coding technique. If a fixed table must be used, we recommend using the table calculated for the sound file PD. That seems to have the least problem with other sound files, although the net reduction is only between 5 and 7 bits per sample. For 16-bit samples, this gives between 9 and 11 bits for each sample for most sounds.

Table III gives the resulting codelengths for the optimal code for sound file PD. The 12-bit number to be coded will be from 0 to 4095. The numbers from 0 to 454 will all have unique codes of 12 bits or less. Above this value, the sample is coded as a unique 12-bit code (there will be one left) followed by the original 12-bit value. Notice that the progression is almost exponential. Each range almost doubles the number of members up to a point. This suggests that some form of floating-point encoding might be useful. We will discuss this further in the next part.

There is a further inconvenience with the use of minimum-length encoding, and that is the problem of error recovery and random access. The problem is that the code is designed to be decoded by examining the stream bit by bit and grouping the variable numbers of bits into samples. If you lose your place in the bit stream, there is no way of identifying the beginning of a new sample. This can be easily corrected by breaking the data into blocks and restarting the code at the beginning of each block. In this case, you must put up with the fact that either each block will be a different length with a constant number of samples per block, or each block will contain a fixed number of bits with a variable number of samples. Moreover, if you wish to reference a particular sample in the middle of a block, you must begin decoding at the beginning of the block.

TABLE I

	LS		PD		TR	
0	9.079/12	10.092/13	9.547/12	11.631/14	11.303/12	9.028/10
1	5.796/12	6.597/13	7.836/12	9.860/14	5.596/12	3.807/10
2	4.966/12	5.899/13	7.767/12	9.795/14	4.836/12	2.973/10

LS: Spoken utterance from male speaker

PD: Computer-generated music with flute, voice, computer sounds, and reverberation

TR: Computer-generated synthetic voice with reverberation

Average code lengths for coding the high-order 10 through 14 bits of the rectified samples for diferent sound files. The number before the slash in each case gives the mean sample length after coding. The number after the slash is the number of high-order bits that were used. The rows marked 0, 1, and 2 denote 0th differences (coding the sample itself), 1st differences, and 2nd differences.

TABLE II

Sounds to be coded:

	DU	SC	FX	FW	TR	PD	LS	
DU	6.069	5.959	5.992	17.320	4.948	8.526	5.044	Table used:
SC	6.697	6.107	6.996	18.158	5.605	9.590	5.790	
FX	6.581	6.536	5.833	19.546	4.974	9.798	5.305	
FW	8.419	8.342	8.593	11.286	8.207	8.952	8.364	
TR	7.037	7.155	6.175	20.605	4.836	10.853	5.397	
PD	6.383	6.201	6.196	14.727	5.552	7.767	5.649	
LS	6.272	6.108	5.969	18.913	4.779	9.375	4.966	

DU: Computer synthesized bell-like tones, no reverberation

SC: Synthetic music instrument tones with reverberation

FX: Live flute in very dry studio

FW: Same flute with concert-hall reverberation

TR, PD, and LS as in table I.

Comparison of mean sample lengths when the code used is computed from the histogram for a different sound file. Each row is a code from the sound file indicated at the left. The columns are the coding of a given sound file by different codes. The diagonal represents the optimal coding. We see that in some cases the sub-optimal coding causes an explosion of the mean sample length.

TABLE III

CODE LENGTH	RANGE
4	0:1
5	2:5
6	6:15
7	16:33
8	34:66
9	67:126
10	127:211
11	212:329
12	330:454
24	All others

Optimum code for sound PD. All numbers between 455 and 4095 are coded as a special 12-bit code (there will be one left) followed by the original 12 bit sample. This keeps a bound on the maximum code length but causes a slight increase in mean sample length.

FLOATING-POINT CODING.

Figure 1 shows the block diagram of the floating-point coder we have been working with recently. Q represents the floating point quantization and coding itself. The inverse of Q represents decoding into integer. In the figure, X is the input signal, Y is the coded output signal, and R is the reconstructed signal at the transmitting end. The receiver need only duplicate the circuitry required to generate R from Y. The coefficient, a, can be thought of as a pre-emphasis quantity. At zero, we have direct floating-point coding, such as is used internally in the converters of Kriz [1975]. At one, we have something resembling ADPCM coding [Jayant 1974, Cummiskey et al 1973] but with the scale factor explicitly transmitted, rather than inferred. This system is a special case of the predictive coding scheme for voice [Atal and Schroeder 1970, 1978, Makhoul and Berouti 1978]. The point of putting the quantization in the loop is so that long-term error does not accumulate. The reconstructed signal will always converge to the desired value in a finite number of samples. This system at a=1 is functionally identical to that described by Samson [1978].

The floating-point coding scheme we are considering represents the integer input sample as a mantissa of b binary bits and an exponent which is almost always 4 bits long. If we force the mantissa to always be normalized, we can save one more bit by throwing out the sign bit, which will always be the complement of the high-order mantissa bit. We will call this process "sign compression." Performing sign compression causes us a problem in representing very small numbers, so we have been using as the exceptional case an exponent of zero to indicate that the mantissa is not sign compressed, but is a two's complement, right-adjusted number. In this case, the sign must be extended to the high order bits. In sign compressed codes (exponent unequal to zero) the complement of the high-order mantissa bit is extended to the high-order bits and the resulting two's complement word is shifted left the number of places indicated by the exponent. The vacated positions are to be filled with zeros. By way of terminology, when we say that a mantissa has b bits, we will mean b bits after sign compression. It would be (b+1) bits if the sign were included explicitly.

The question is now what values of the numbers b and a provide the greatest compression with the least amount of perceptable noise? For a given setting of b and a, the total dynamic range representable is determined by the number of choices of shift. For this study, we shall insist on at least a 16-bit dynamic range, with further dynamic range considered to be icing on the cake, i.e., desirable but not absolutely essential at this time.

As a step toward understanding the behavior of the coding algorithm, we performed a series of simulations of the system on a general-purpose computer with varying values of the parameters. The error was computed as follows:

$$(1) \qquad E = 10 \log_{10} \frac{N \, \Sigma \epsilon(n)^2 - (\Sigma \epsilon(n))^2}{N \, \Sigma X(n)^2 - (\Sigma X(n))^2}$$

where X(n) is the input signal at time nT, T being the sampling period. $\epsilon(n)$ is the error signal, which is just (R(n)-X(n)), where R(n) is the reconstructed signal.

This error measure gives us the expected results when the coefficient a is set to zero. For pure floating point, the error is approximated by the following:

$$(2) \qquad E_{a=0} = 6(b+1.5)$$

We have added 1.5 to the number of bits in the mantissa for two reasons. The first is that sign compression was used universally in this study which adds one bit to the effective mantissa length. The second is that rounding the signal before coding always reduced the noise level by 3dB, so that rounding before quantizing is also used universally in this study. For b=8, the error using different values of the coefficient a is shown in figure 2. The inputs were pure sinusoids of full amplitude. We can thus expect that b=8 gives about 57 dB of signal-to-noise ratio (for a=0), 10 bits gives about 69 dB, and so on. The curves in figure 2 are only given for b=8 because other values of b only shifted the curves up or down by 6 dB per bit. The shapes were identical. The "bumpy" character of the a=1 curve shows in a very evident way the changing of scale as the slopes of the sinusoids grow with increasing frequency. One important thing to notice is that in no case does the maximum error exceed the error for the pure floating point case (a=0) by more than 6 dB, and for frequencies less than .16 of the sampling rate, every increase of the coefficient reduced the error. Any value of a above 1.0 drives the reconstruction filter unstable and is thus considered undesirable.

All this discussion of the coding scheme and its error may be interesting but not terribly useful until we determine its relation to perception. At first glance, the increasing error with increasing frequency may seem detrimental. It is not clear either that sinusoids form a representative test set, that one can generalize from pure sinusoids to musical sound. As an illustration, one can site the case of transient intermodulation distortion in amplifiers, where an amplifier can function perfectly with sinusoids but distort greatly for high slew-rate transient signals. Likewise, since the coding/decoding process is highly nonlinear, we might expect that the error could increase when transmitting more complex spectra than simple sums of sinusoids.

We tried several different kinds of filters for coding and reconstruction besides the simple 1st order system shown here, but every attempt gave no appreciable gain in overall precision. We could trade distortion in one frequency region for distortion in another frequency region, but were unable to find another kind of filter that uniformly reduced the distortion as did the simple 1st order filter shown. This filter has the further advantage that it can be realized entirely without multiplications. Realization of the floating-point coder and decoder requires nothing more than binary shift.

RELATION TO PERCEPTION

To begin the study of the relation of figure 2 to perception, we did informal tests, both physical, gedanken, and simulated, on the effects of this kind of coding. What we seemed to be perceiving was that the extremes of frequency were much more sensitive to distortion than the mid frequencies. The lows expecially were extremely sensitive. We looked at the error spectra to try to find some clue as to why this might be the case, but the error spectra were uniformly flat [Bennet 1948]. They looked much like white noise spectra. Figure 4 shows error spectra for seven different frequencies of sinusoids coded and decoded via this method. For the highest frequencies, or for frequencies very close to some integral divisor of the sampling rate, there were sinusoidal distortion products, but they were again distributed uniformly throughout the spectrum. They did not bunch or cluster in particular spectral regions. This led us to believe that this difference in sensitivity could only be the result of a perceptual phenomenon.

To determine, at least in the steady state, whether a given sound will be perceptable in the presence of another sound, we can use the results on loudness summation in the presence of masking. This exposition will follow the theories of Zwicker and Scharf [Zwicker 1958, Zwicker and Scharf 1965]. Since their theory is much too complex to give in full detail here, we will only attempt to highlight certain of the features that are most relevant to our discussion of perceptability of quantizing noise.

To form the loudness estimate of a sound, we start with the energy in each critical band. For each critical band, the appropriate masking pattern is selected according to the frequency and total energy in the band [Zwicker 1958]. The masking pattern itself is taken to be representative of the excitation pattern on the basilar membrane. The specific loudness is then calculated from the masking patterns for each band by a modified power law. We then superimpose all the patterns for all the critical bands. These patterns define an upper envelope. The integral of the specific loudness between the threshold curve and this envelope is then the loudness.

What this implies is that if we then add any other sound, if it does not rise above this upper envelope, it will not be heard. We then should be able to predict from the published masking patterns the perceptibility of the quantization error. We implemented this model in software to attempt to predict loudness and audibility [Moorer 1975, Grey and Gordon 1978]. The results were that the model could indeed be adjusted to give measures of loudness that were consistant with experimental évidence, but the prediction of audibility or non-audibility was far too sensitive to the exact shapes of the masking patterns and the threshold curve to be reliable.

Happily, even in the absence of a rigorous and precise theory for this question, we do have two experimental works along this line. We will take the approach of Lee and Lipschutz [1972] in that we would like that the signal mask the quantization error under all circumstances. Since at lower frequencies the error of this coding scheme behaves much like white noise, we can appeal to psychoacoustic data concerning the masking of white noise by various signals [Young and Wenner 1967]. Their principal results are the following:

(1) White noise must be of intensity greater than 16 dB SPL to be heard at all. This is called the "unmasked" noise threshold.

(2) The threshold of perceptibility of white noise was not affected by the presence of sinusoids of any frequency when their amplitudes were less than 80 dB SPL.

(3) Sinusoids between 90 and 120 dB SPL produced a great variation in the threshold of perceptibility of white noise as a function of the frequency of the sinusoid. The threshold was uniformly raised by the presence of the sinusoid, indicating that the noise had to be at a greater amplitude to be heard.

(4) Sinusoids in the range of 700 to 1000 Hz demonstrated the greatest masking of white noise.

(5) Increasing the harmonic content of the signal only increased the amount of masking.

In other words, very low and very high tones do not mask the white noise hardly at all, whereas sinusoids in the central frequency range mask the white noise considerably. One can see that this corresponds with what we might predict from the models of Zwicker [1958]. Figure 3 shows a schematic representation of the masking patterns at three different frequencies. The U-shaped curve at the bottom is a stylized representation of the threshold of hearing. The dotted line is the excitation pattern for pure white noise. The frequency scale is in Barks which is a transformation based on critical bandwidths and also on distance along the basilar membrane. We can see that the masking pattern for the intermediate frequency covers (masks) a larger proportion of the white noise pattern than does either the very low frequency or the very high frequency and is thus generally consistant with the results of Young and Wenner.

To test the relation between these findings and our own case, we performed experiments to determine the perceptibility of the quantization distortion with our coding scheme. It quickly became apparent that no other value of the coefficient, a, was useful except the value 1.0. The reason for this is that the audibility of the quantization noise is much increased at low frequencies which implies that much more precision is necessary. All other values sacrifice precision in the lows for little gain in the highs. The complete range from a=0 to a=1 only changes the distortion for the highs by 6dB, whereas it reduces the distortion for the lows by more than 35 dB. This line of reasoning also implies that it is never advantageous to use straight floating point (a=0). It is virtually always better to use differential floating point (a=1) for coding audio.

By informal listening tests, we narrowed down the ranges of bits for each frequency and decided on the following test paradigm: seven frequencies were chosen. At each frequency, four mantissa lengths were chosen such that the lowest number of bits clearly demonstrated audible distortion and the highest number of bits was indistinguishable. We then presented the uncoded and coded signals in groups of three, two uncoded and one coded, and asked the subject to write down which of the three sounded different from the other two. If the subject is guessing, we would expect a probability of 1/3 for each category. Every trial was presented three times with the coded signal in position 1, 2, and 3, and the order of presentation was completely randomized. Since we had recorded the stimulii on magnetic tape using dBx noise reduction, we found that the imperfections on the tape (dropout) causes experimental bias. We produced three different tapes with three different orders in hopes that the bias would average out. The stimulii were presented to the subjects by loudspeakers (JBL 4343 studio monitors) in a relatively dry room with a background noise level (including the sound of the tape recorder) of 26 dB SPL (A-weighted). This experiment can not be considered absolutely definitive because of the fact that the stimulii were recorded on magnetic tape rather than presented directly from the computer, but this gives us some reasonable guidelines toward chosing a mantissa length.

The results of the experiment are shown in table IV and figure 5. In Table IV, the first column gives the frequency of the sinusoid, and the second column gives the ratio of that frequency to the sampling rate which was 25600 Hz. The third column is the resulting sound intensity at the subject in dB SPL. There is variation due to the (uncompensated) irregularity of the responses of the loudspeakers. For each frequency, as the number of bits increases, we expect the fraction of correctly identified coded signals to drop to the guess level, which is 1/3. We fit the subjects' responses with the following two-parameter sigmoid function:

$$(3) \quad f(b) = 1 - \frac{2}{3}\,\frac{1}{1 + e^{-\beta(b-\alpha)}}$$

The number of bits at the 95% confidence level (when the sigmoid function dropped below 0.367) is shown in the fourth column in table IV, and the fifth column gives the error in dB at that number of bits. The sixth column gives the predicted threshold from the Young and Wenner [1967] data. In figure 5, the subject's responses in terms of the fraction of correct responses is plotted versus the number of bits in the mantissa of the coded-decode signal. The vertical extent of the cross represents a deviation of plus and minus one root-mean-square error from the fitting of these data by the sigmoid function. The horizontal extent of the cross represents this same RMS error divided by the slope of the sigmoid function at that point (limited to plus or minus one bit maximum). The 95% confidence level is represented by a small square along the sigmoid function, and the deviation of one RMS error unit from the random level of 1/3 is represented by a small triangle. Most of the time, these levels coincide.

These results confirm our prediction that the noise is more perceptible at the lower frequencies, and reasonable correspondance is shown between our results and those of Young and Wenner. At higher frequencies, our results diverge from those of Young and Wenner. Possibly this is due to the fact that the form of the distortion is no longer like white noise at these higher frequencies. In any case, this should be a subject of further study.

From these results, we can safely say that for sinusoids of all but the highest frequencies, a mantissa length of 9 bits should be sufficient. At higher sampling rates, we can expect that even fewer bits would be needed because figure 2 shows that doubling the sampling rate decreases the error almost universally by 6 dB.

TABLE IV

FREQUENCY		dB SPL	av bits	error	predicted
60.3	.0023	105	6.15	-82.2	-89.0
139.6	.0054	100	7.31	-80.0	-84.0
323	.0126	96	8.91	-82.5	-80.0
747.5	.0292	96	8.44	-72.9	-66.0
1729	.0676	93	8.63	-67.4	-66.0
4003	.1564	92	7.00	-51.4	-76.0
9263	.3619	88	9.55	-60.8	-72.0

Results of perceptibility experiment for the floating-point coding scheme with a=1. The sampling rate was 25600 Hertz. For each of the seven frequencies, we list the frequency in Hertz, the ratio of the frequency to the sampling rate, the resulting intensity in dB SPL (which varied due to unevenness in the speaker response), the average number of mantissa bits required for 95% certainty of indistinguishability, and the resulting error at that number of bits. The sixth column shows the audibility of white noise as derived from the Young and Wenner [1967] data.

DISCUSSION

These data show that for sinusoidal signals of normal range, no more than 9 bits of mantissa are necessary (if sign compression and rounding are used), giving a 13-bit sample. This is a net savings of 3 bits per sample, which while not striking in itself is capable of a dynamic range equivalent to a 24-bit integer sample. In any case, any savings at all will be greatly appreciated by the digital tape recorder manufacturers, in that they are already up against the problem of the tremendous bit densities required. This is also for highest quality reproduction. For lesser quality, such as home recordings or other secondary channel uses, a smaller 11 or 12 bit sample could be envisioned which still preserves the dynamic range of a much larger sample.

Notice that this implies that under certain circumstances, the 12-bit straight floating-point schemes of Kriz [1975] and Blesser [1975] will exhibit perceptible distortion, especially for very low tones. Happily, low-frequency sinusoids virtually never occur in nature. It is most likely that one could only show up the problem with computer-generated tones. Indeed, even the 14-bit straight PCM system used by the BBC may demonstrate some audible distortion on certain low tones of high spectral purity.

The floating-point incremental scheme described has a constant number of bits per sample, unlike the minimum-length scheme first described, but shares with minimum-length encoding the property of being non-restartable. That is, the entire sound file must be decoded from beginning to end: one cannot conveniently start in the middle. For this reason, we recommend storage by blocks where the first, say, 24 bits of the block is the exact integer value of the previous sample of the last block, and the remainder of the block is then the differences as previously described. This allows one to restart on any block boundary, but still requires reading through the block to restart at a given sample within a block.

As for hardware realizations of this scheme, one might be tempted to realize the decoding algorithm with a low-order (10-bit, for instance) converter for the mantissa followed by scale change to effect the exponent and an integrator to realize the summation. It is possible that this could be made to work, but the fundamental difficulty is that the frequency response of a digital summation is different from that of an analog integrator. Some compensation filter would have to be included to flatten the overall response.

There remain still a few points to discuss. One objection might be phrased as the following question: Why should the results for the sinusoids be taken as the worst-case? Why shouldn't there be some other signal that would show up the error more easily than the sinusoid? This is easily answered in that the sinusoid shows the minimum masking for the given loudness. All other signals exhibit a greater degree of masking, and thus will certainly reduce the perceptibility of any distortion present. The higher the harmonic content of a signal, the more complete the masking of the quantization will be. One might then ask about transients. The fact that this system takes several samples to converge to a given value, doesn't that mean that there will be distortion of the transients? Indeed it does, but likewise, a transient has a very large bandwidth, and thus relatively high masking capabilities.

Another question might be whether the high level of distortion on the high frequencies is acceptable or not. The fact that the distortion increases to a maximum at about .23 of the sampling rate and that this maximum is 6 dB higher than the comparable straight floating-point system (a=0) has led some authors to regard this method as "totally unacceptable" [Blesser 1978, p 752]. We cannot entirely understand the harshness of this complaint in that the only signals that can occur as full-amplitude sinusoids of these frequencies are machine-generated signals. For digital recording studio purposes, using live or natural sources, full-amplitude high-frequency signals simply do not occur. As soon as the sinusoid is not perfectly pure (harmonic distortion down more than 50 dB), the extra masking allows a much higher level of noise to be present. If one were only trying to decide between straight floating point and incremental floating point, there should be no question, because for the same sample length in bits, the increase in fidelity for the low tones of any amplitude is much more important than the loss of fidelity in high-amplitude highs. If one were trying to decide between straight PCM and incremental floating point, one would have to consider carefully the nature of the audio material to be recorded. Although the computer is perfectly capable of generating such high-amplitude sinusoids, it is not necessarily what computer music composers wish to do all the time. In fact, our experience over ten years of computer music work at Stanford suggests that this occurs in practice very little. For a definitive system, however, one might reasonably "hedge one's bets" by use of a dual-mode system, such that a bit is carried along at the beginning of each block giving the coding of that block, be it straight PCM or incremental floating point or whatever. It would then be the responsibility of the operator at the time the recording is made to chose the coding technique he feels will give the best results for the given sound.

The only reason not to use a coding scheme such as the one described would be if one were going to do further processing on the signal. For instance, filtering the signal such as to amplify some part of the spectrum that is not occupied by signal would have the effect of amplifying the noise in that spectral region. If the filtering were strong enough, this could indeed amplify the noise to the point where it would be audible. Likewise, if one intended to do mathematical analysis on the signal to, for example, extract physical parameters of musical instrument tones, or to compare distortion figures of amplifiers or something, one might indeed want precision beyond that of the sensitivity of human hearing.

CONCLUSIONS

Minimum length encoding and incremental floating-point encoding were considered as possible schemes for reducing the amount of data in high-quality digitized audio. It was found that optimum coding on the second differences of the signal produced the greatest reduction of data, but that certain signals could only be reduced very slightly. Furthermore, if a coding table is used that does not correspond to the optimum for the sound, an explosion of the amount of data up to a factor of two over the original rate can occur. Despite this, our experience suggests that for the most part, reductions of 5 to 7 bits per sample can be expected. The inconvenience of the method is that storage in blocks of fixed numbers of bits results in each block possibly containing a different number of samples, a fact which may or may not be a problem, depending on the application.

For floating-point incremental coding, a psychoacoustic experiment was performed to determine the perceptibility of the error. It was found that with pure sinusoids, a mantissa of 9 bits or more assured indiscriminibility at the 95% confidence level for all but the highest frequency tones. This number assumes sign compression and rounding has been performed in the floating-point conversion. This provides a uniform reduction of 3 bits per sample for highest-quality results and even more if ultimate quality is not necessarily the goal. This method has the advantage that the number of bits in each sample is constant, but shares with minimum-length encoding the disadvantage that a sample in the middle of a data block cannot be referenced at random but the block must be read from the begining. Furthermore, appeal was made to the theory of masking to support the claim that testing with pure computer-generated sinusoids is the worst case and that with any normal musical sound, the coding error will be even more completely masked.

In conclusion, there seems to be little reason to use straight PCM coding for digital storage and transmission of audio except when extremes of precision and predictable error characteristics are needed.

FIGURES

1) Block diagram of coding scheme. Q represents floating-point encoding of the signal and Q↑-1 represents decoding of the signal. The coefficient, a, is set to zero for straight floating-point coding and to 1.0 for differential coding.

2) Error rates for an 8-bit mantissa with sign compression and rounding for different values of the coefficient, a. The input in this case were pure sinusoids of full amplitude.

3) Masking patterns for three different frequencies. The U-shaped curve on the bottom is the threshold of hearing. The dotted line across the bottom is the power spectrum on a Bark scale of uniform white noise.

4) Error spectra for floating-point incremental coding at 7 different frequencies. The mantissa length was uniformly 6 bits. The coefficient, a, was set to 1. The frequencies in terms of fractions of the sampling rate were (a) .0023, (b) .0054, (c) .0126, (d) .0292, (e) .0676, (f) .1564, and (g) .3619. Note that in most cases, the signal resembles white noise very closely. In two cases, (e) and (g), there is a marked non-white nature, but the error is broadband.

5) Results of discrimination experiment. The frequencies are the same as shown in Table IV and in figure 4. The four crosses in each figure mark the fraction correct for discrimination at each of four mantissa lengths. The smooth curve is a sigmoid function that was fit to the four points my minimizing the mean square error. The vertical strokes in each cross indicate deviation by plus and minus one RMS error. The horizontal strokes are the RMS error divided by the slope of the sigmoid curve at that point to give a rough indication of the variability in terms of number of bits. The square marks the 95% confidence level, the triangle marks one RMS error from 1/3.

REFERENCES

H.G. Alles, P. diGiugno "A One-Card 64-Channel Digital Synthesizer," Computr Music Journal, Volume 1, Number 4, November 1977, pp7-9

S. Alonso, J.H. Appleton, C. Jones "A Special Purpose Digital System for Musical Instruction, Composition, and Performance," Computers and the Humanities, Volume 10, pp209-215

B.S. Atal, M.R. Schroeder "Adaptive Predictive Coding of Speech Signals," Bell Systems Technical Journal, Volume 49, October 1970

B.S. Atal, M.R. Schroeder "Predictive Coding of Speech Signals and Subjective Error Criteria," in Proceedings of the 1978 IEEE conference on Audio, Speech, and Signal Processing, 1978, p573

W.R. Bennett "Spectra of Quantized Signals," Bell System Technical Journal, Volume 27, 1948, pp446-472

B.A. Blesser "Digitization of Audio: A Comprehensive Examination of Theory, Implementation, and Current Practice," Volume 26, Number 10, October 1978, pp739-771

B.A. Blesser, K. Baeder, R. Zaorski "A Real-Time Digital Computer for Simulating Audio Systems," Journal of the Audio Engineering Society, Volume 23, Number 9, November 1975, pp698-707

P. Cummiskey, N.S. Jayant, J.L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech," Bell Systems Technical Journal, September 1973, pp1105-1118

J.M. Grey, J.W. Gordon, "Perceptual Effects of Spectral Modifications on Musical Timbres," Journal of the Acoustical Society of America, 1978

D.A. Huffman "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the I.R.E., Volume 40, September 1952, pp1098-1101

N.S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," Proceedings of the IEEE, Volume 62, Number 5, May 1974, pp611-632

J.S. Kriz "A 16-bit A-D-A Conversion System for High-Fidelity Audio Research," IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume ASSP-23, Number 1, February 1975, pp146-149

F.F. Lee, D. Lipschutz "Floating Point Encoding for Transcription of High Fidelity Audio Signals," presented at the 55th convention of the Audio Engineering Society, October 1976, preprint 1190 (L-1)

J. Makhoul, M. Berouti "High Quality Adaptive Predictive Coding of Speech," in Proceedings of the 1978 IEEE conference on Audio, Speech, and Signal Processing, 1978, pp303-306

M.V. Mathews "The Technology of Computer Music," MIT Press, Boston, Massachusetts, 1969

J.A. McCracken "A High-Performance Digital Audio Recoder," Journal of the Audio Engineering Society, Volume 26, Number 7/8, July/August 1978, pp560-562

J.A. Moorer, "On the Loudness of Complex, Time-Variant Tones," Stanford University Music Department Memo STAN-M-4, 1975, 18pp

J.A. Moorer, "Signal Processing Aspects of Computer Music: A Survey," Proceedings of the IEEE, Volume 65, Number 8, August 1977, pp1108-1137

J.P. Myers, A. Feinberg "High Quality Professional Recording Using New Digital Techniques," Journal of the Audio Engineering Society, Volume 20, Number 8, October 1972, pp622-628

P.R. Samson "Incremental Floating-Point Coding," Journal of the Audio Engineering Society, Volume 26, Number 7/8, July/August 1978, pp556-558

R.B. Warnock "Longitudinal Digital Recording of Audio," Audio Engineering Society Preprint number 1169 (L-3), 55th Convention, October 29 to November 1, 1976

I.M. Young, C.H. Wenner "Masking of White Noise by Pure Tone, Frequency-Modulated Tone, and Narrow-Band Noise," Journal of the Acoustical Society of America, Volume 41, Number 3, 1967, pp700-706

E. Zwicker, B. Scharf, "A Model of Loudness Summation," Psychological Review, Volume 72, Number 1, 1965, pp3-26

E. Zwicker, "Ueber Psychologische Und Methodische Grundlagen Der Lautheit," Acustica, Volume 8, Number 1, 1958, pp237-253

fig 1

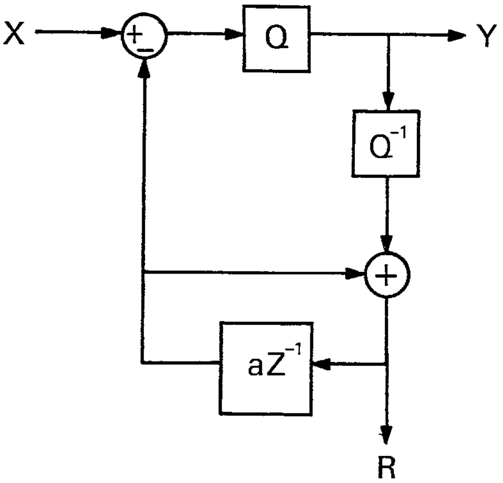

X
+
–
Q
Y
Q^{-1}
+
aZ^{-1}
R

fig 2

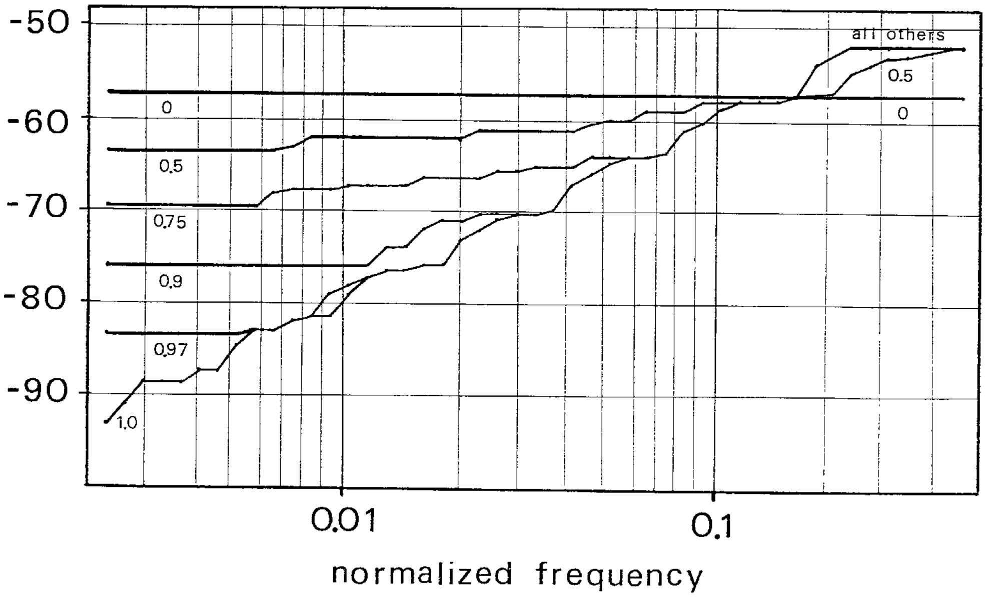

-50
-60
-70
-80
-90
0.01
0.1
normalized frequency
all others
0.5
0
0
0.5
0.75
0.9
0.97
1.0

fig 3

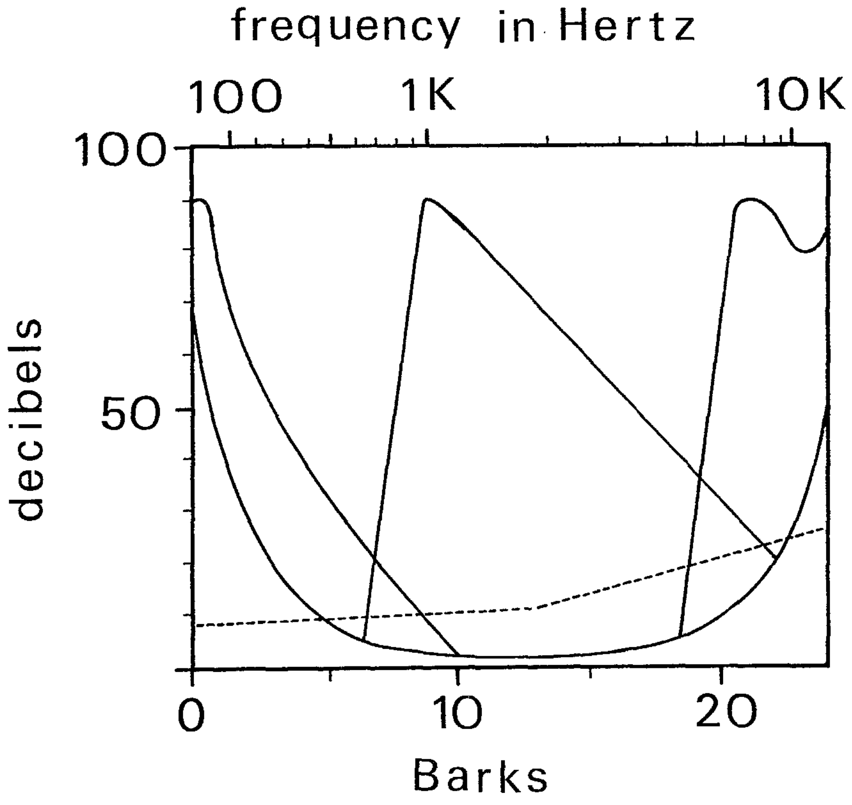


fig 4a

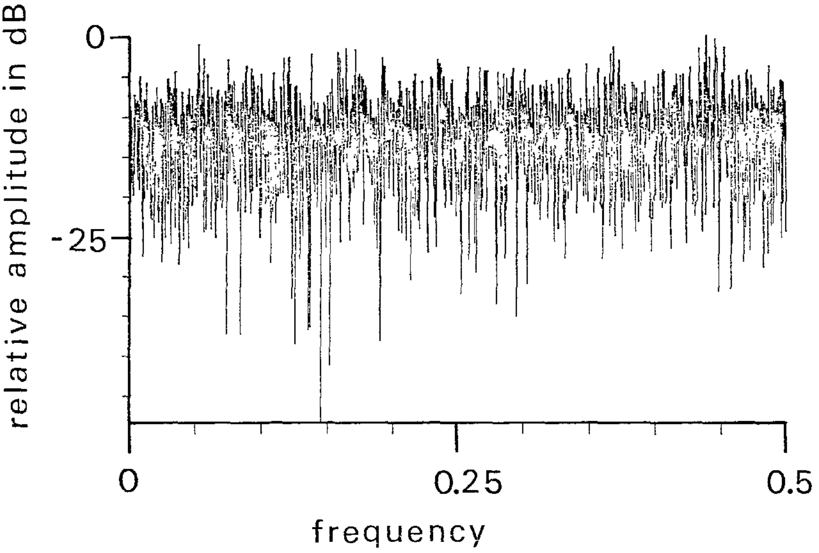

relative amplitude in dB
0
-25
0
0.25
0.5
frequency

fig 4b

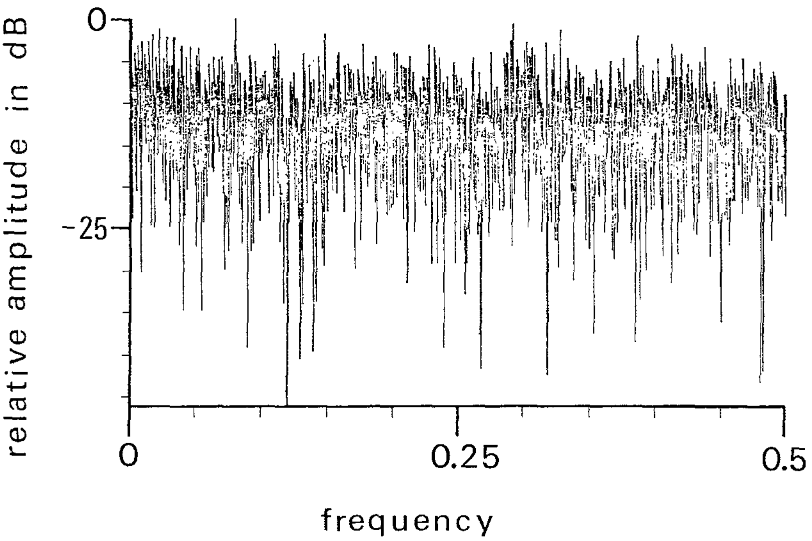
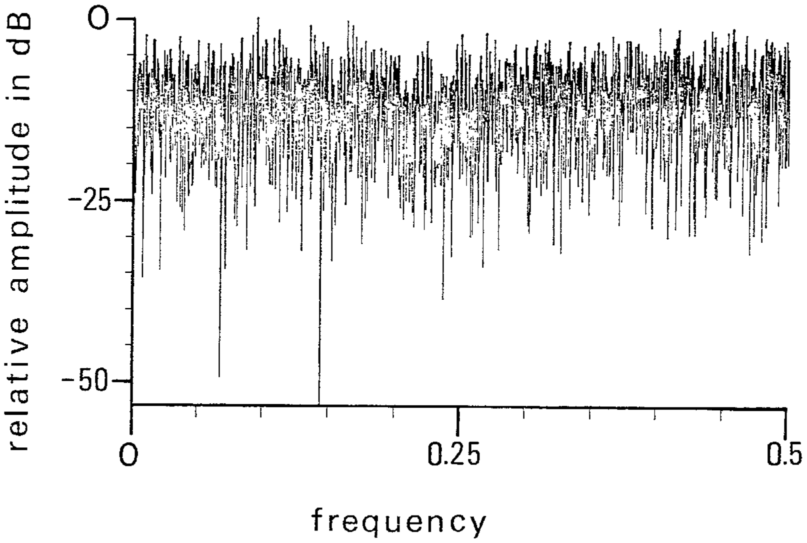
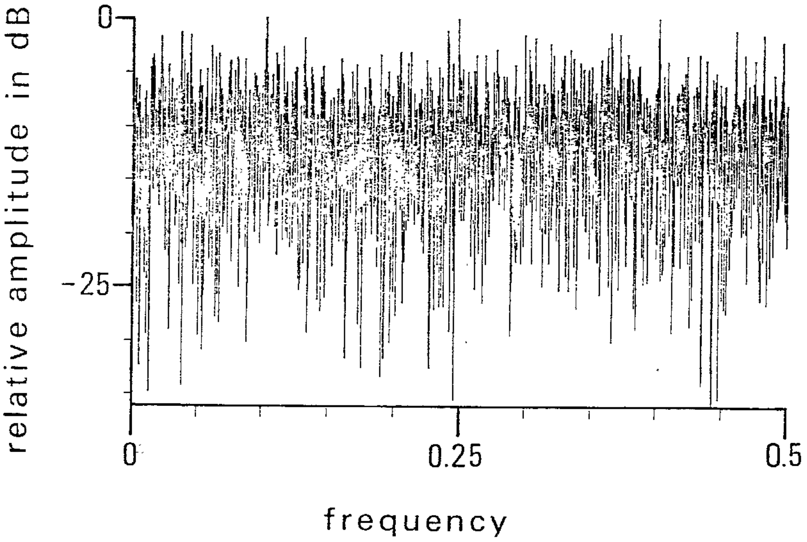

relative amplitude in dB
0
-25
0
0.25
0.5
frequency

fig 4c



relative amplitude in dB
0
-25
-50
0
0.25
0.5
frequency

fig 4d



relative amplitude in dB
0
-25
0
0.25
0.5
frequency

fig 4e

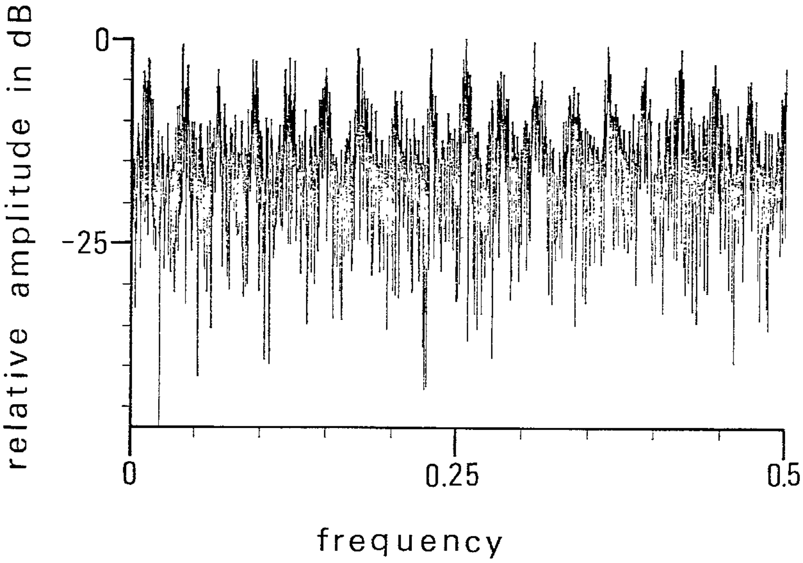

relative amplitude in dB
0
-25
0
0.25
0.5
frequency

fig 4f

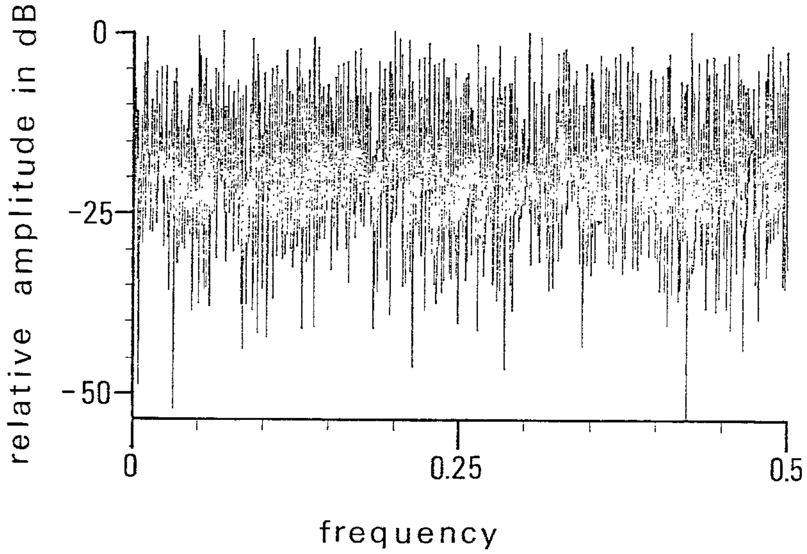
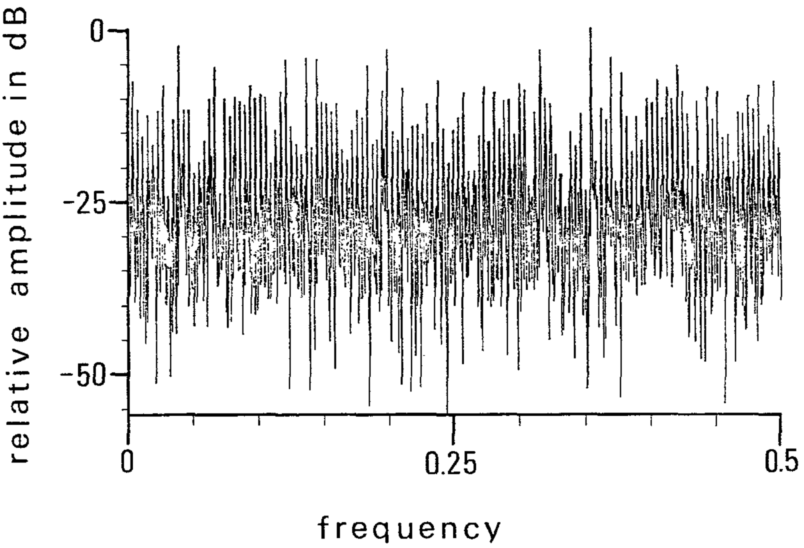

relative amplitude in dB
0
-25
-50
0
0.25
0.5
frequency

fig 4g


relative amplitude in dB
0
-25
-50
0
0.25
0.5
frequency

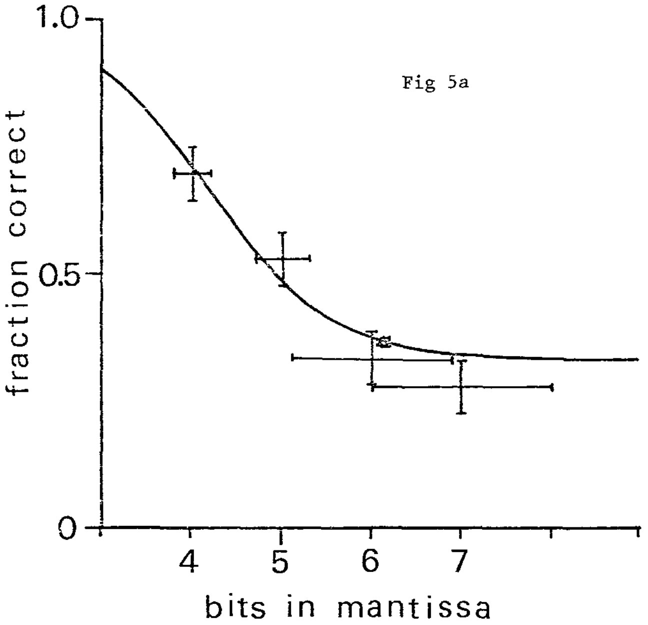


Fig 5a

fraction correct
1.0
0.5
0
4
5
6
7
bits in mantissa

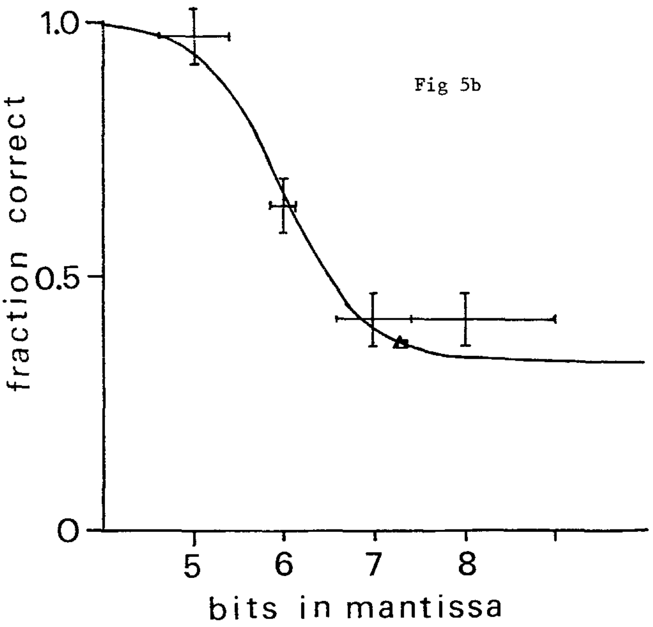

1.0
0.5
0
fraction correct
5
6
7
8
bits in mantissa

Fig 5b

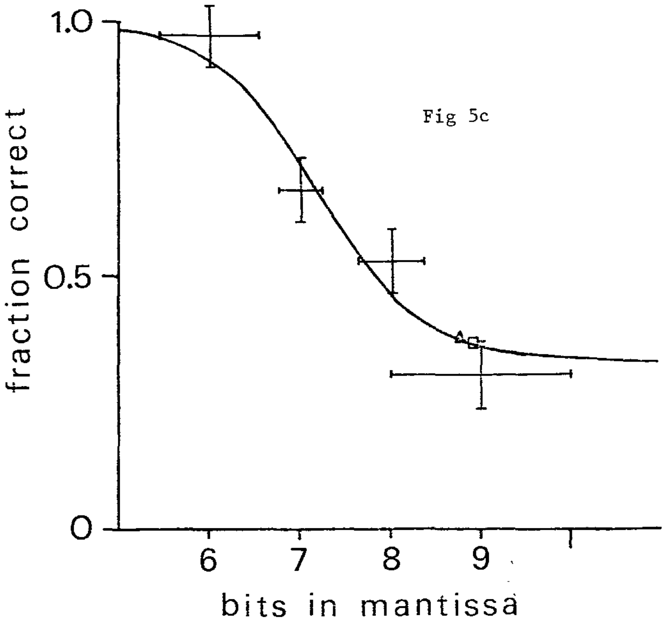

1.0
0.5
0
fraction correct
6
7
8
9
bits in mantissa

Fig 5c

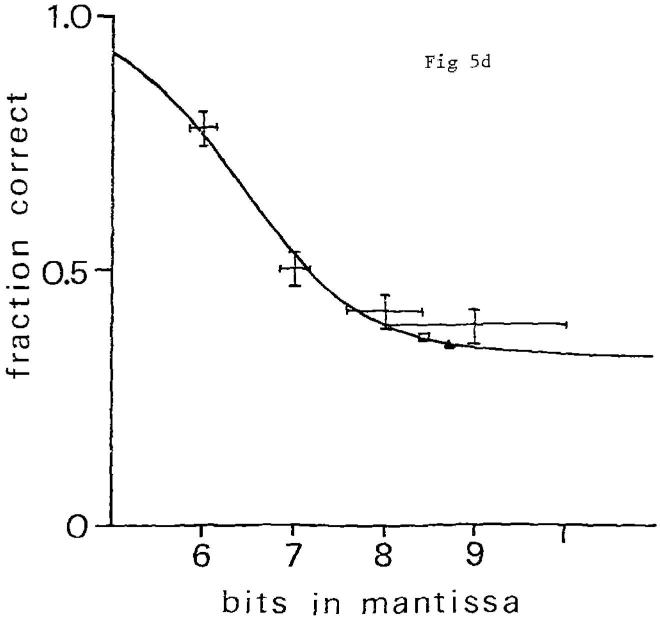


Fig 5d

1.0
0.5
0
fraction correct
6
7
8
9
bits in mantissa

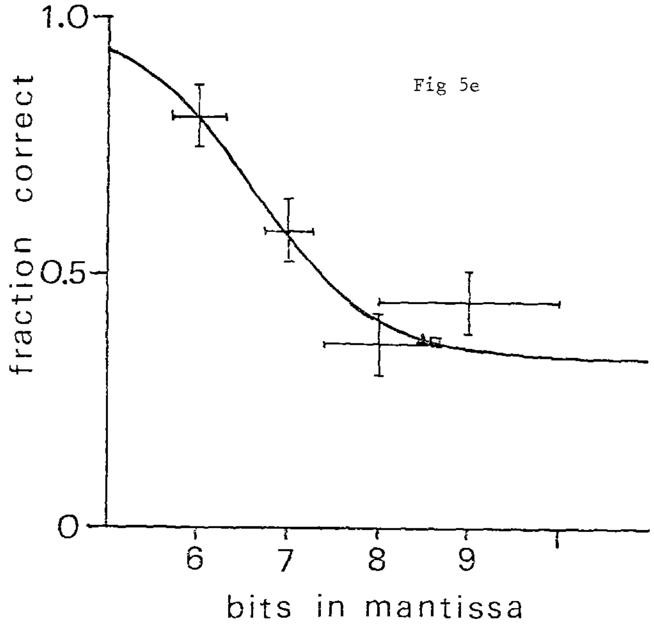


Fig 5e

1.0
0.5
0
fraction correct
6
7
8
9
bits in mantissa

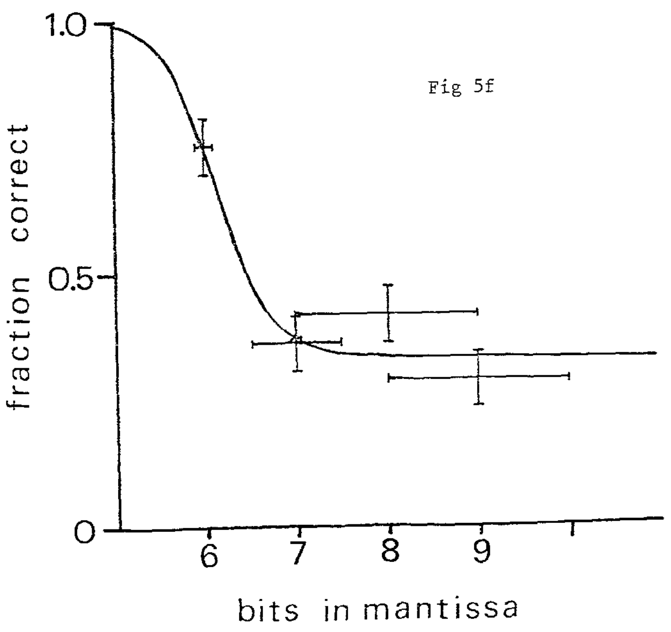

1.0
0.5
0
fraction correct
6
7
8
9
bits in mantissa

Fig 5f

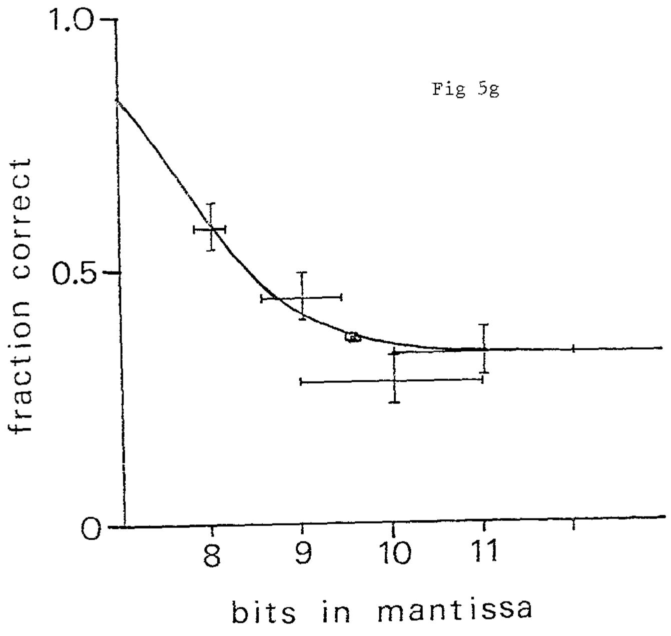

1.0
0.5
0
fraction correct
8
9
10
11
bits in mantissa

Fig 5g