

APT-X100: A LOW-DELAY, LOW BIT-RATE, SUB-BAND ADPCM AUDIO CODER FOR BROADCASTING

MIKE SMYTH
STEPHEN SMYTH

Audio Processing Technology Ltd
Belfast, BT9 5AF, N. Ireland

Details are presented of the apt-X100 16-bit to 4-bit digital audio compression algorithm for real-time transmission of very high quality audio at 128kbits/s. Designed to directly replace digital PCM hi-fi audio links, apt-X100 is implemented on a single masked-ROM DSP and boasts exceptional tolerance to bit errors, low coding delay, zero-overhead automatic data synchronisation and de-multiplexing, and an auxiliary data transmission facility. A complete half-duplex dual-channel encode/decode implementation within a single DSP is also reported.

INTRODUCTION

Consumer expectation of broadcast audio quality has risen over recent years and been largely satisfied with, for example, NICAM coding for domestic television programmes and the replacement of AM with stereo FM radio transmissions. Since the launch of the Compact Disc digital audio has become largely synonymous with high quality audio, but its transmission and storage presents some difficulties to broadcasters due to the large and expensive bandwidth required to provide such services. Nevertheless, significant advances have been made in recent years, particularly in digital signal processing hardware, enabling more complex, higher performance coders to be realised which allow for sizeable reductions in the digital audio data rate. This reduction has allowed digital audio to be progressively introduced as a replacement for weak analogue links in the studio environment.

This paper outlines the important techniques used in the apt-X100 coding system, and briefly describes hardware currently available to facilitate its widespread use. The performance of apt-X100 is indirectly compared to transform-based coders on the basis of noise masking thresholds, and an illustrative comparison with NICAM is also presented.

The apt-X100 audio coding system is currently in use worldwide in studio-transmitter microwave links, DBS radio, satellite-based outside broadcast links, ISDN commentary links and additionally in numerous audio storage products such as floppy-disk based digital audio recorders.

1. APT-X100

The apt-X100 algorithm aims to code transparently and in real-time very high quality digital audio signals at 4 bits per sample. The three key components of the algorithm which collectively achieve this are sub-band coding, linear prediction and adaptive quantisation.

1.1 Sub-band coding

Sub-band coding splits up the audio signal into a number of frequency bands. By doing so the frequency domain redundancies within the audio signals can be exploited, allowing for a reduction in the coded bit rate compared to PCM for a given signal fidelity. The spectral redundancies are due to the signal energies in the various frequency bands being unequal at any instant in time. By altering the short-term coding resolution in each band according to the energy of the sub-band signal, the quantisation noise can be reduced across all bands over a PCM coder for the same overall bit rate. A mechanism must also be provided for allocating bits to each band in proportion to the signal variance within that band.

On its own, sub-band coding incorporating PCM in each band, is capable of providing a gain over full-band PCM for stationary 'non-flat' input signals, gain being defined as the ratio of the quantisation error variances at the same transmission rate. This gain increases with the number of bands, and with the extent of the 'non-flatness' of the input signal, that is with the ratio of the arithmetic mean to the geometric mean of the sub-band variances. Figure 1

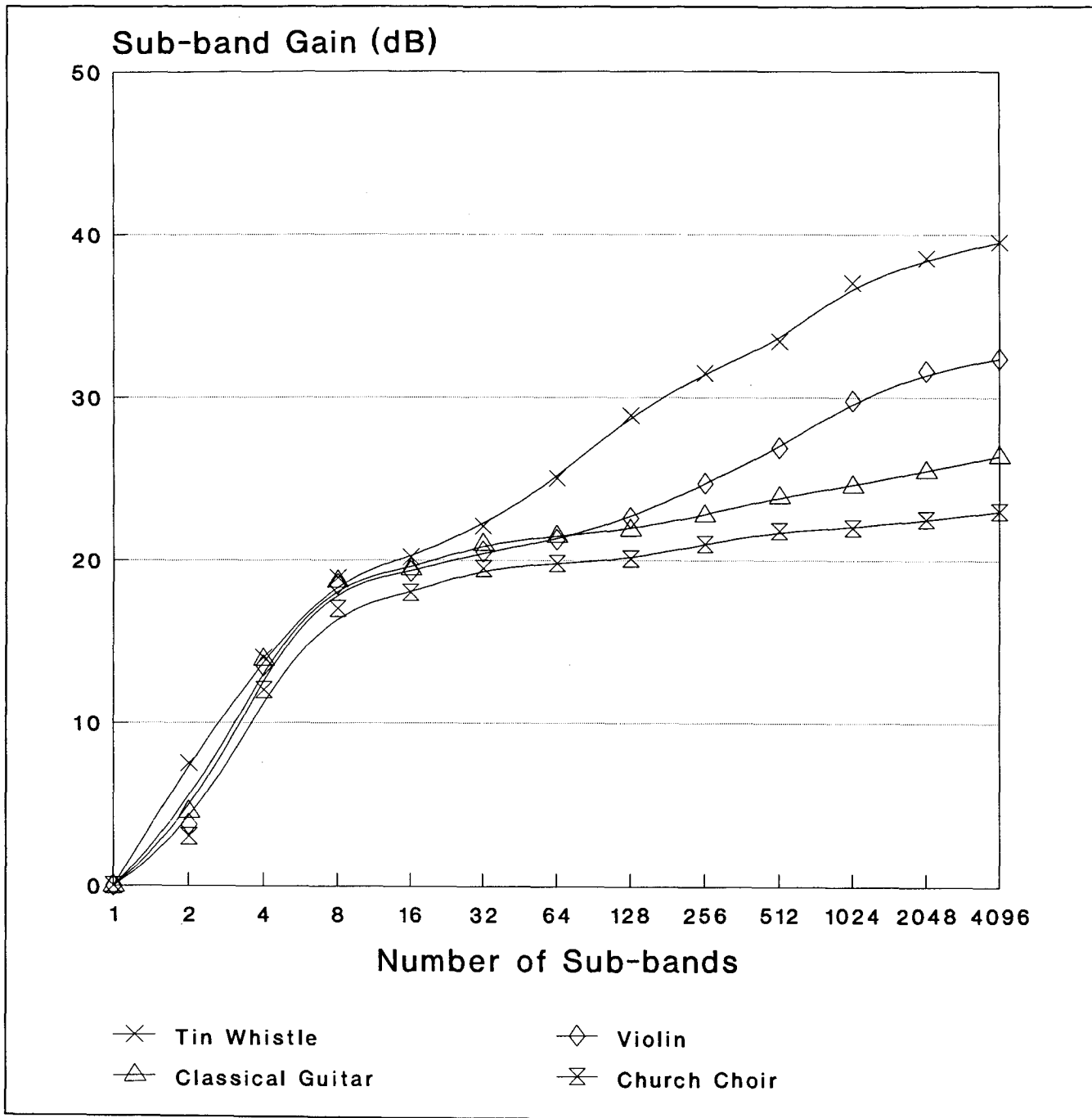


Figure 1 Variation of sub-band gain with number of sub-bands

illustrates the variation of sub-band gain with number of bands for four essentially stationary audio signals.

In practical implementations of sub-band coders, two factors tend to limit the number of bands employed. First, the non-stationarity of audio signals leads to an averaging of the energy across bands, reducing the arithmetic mean to geometric mean ratios, and hence reducing the coding gain. Second, the transmission of the bit-allocations for each of the sub-bands becomes problematical. The key issue in the analysis of a sub-band framework is in determining the likely sub-band gain associated with high quality audio and in

determining the relationships between the sub-band gain, the number of sub-bands and the response of the filter bank used to create the sub-bands. The coding delay and computational complexity are two further problems that must be addressed, both of which tend to limit the number of sub-bands used in a real-time coder.

The apt-X100 algorithm use Quadrature Mirror Filters (QMFs) to divide the input data into four uniform frequency bands. These allow for the exact reconstruction of the audio to within the level of quantisation noise introduced, and also exhibit a constant coding delay across their frequency range. To reduce inter-band leakage,

which reduces sub-band gain, the number of filter taps are weighted to the first QMF stage. With just four bands and a 64/32 tap split, the complexity and delay time of the filter is manageable, whilst the sub-band gain is still considerable.

1.2 Linear prediction

Any reduction in potential gain from the use of just four sub-bands can be compensated for by linear prediction, which aims to remove spectral redundancies which remain in the sub-bands. The total performance of the coder with the inclusion of linear prediction can easily match the sub-band gain associated with the use of many more sub-bands, without resorting to an increased filter complexity or coding delay.

Redundancy is removed by subtracting from the incoming sub-band signal a predicted signal, creating a difference signal which may then be quantised. If the prediction is accurate then the magnitude of the difference signal will be less than the original sub-band signal, allowing the quantisation resolution to be reduced. The degree to which the sub-band signal is attenuated is termed the prediction gain and represents an improvement in the coded SNR over PCM. What is removed at the encoder is added back onto the quantised difference signal received at the decoder.

The apt-X100 algorithm utilises backward adaption of the predictor coefficients which involves no

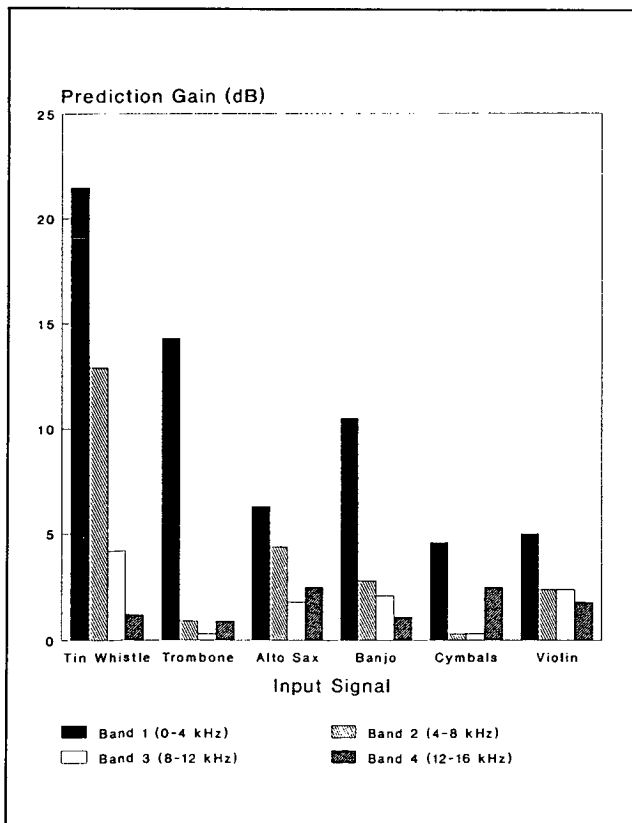


Figure 2 Sub-band prediction gains (apt-X100)

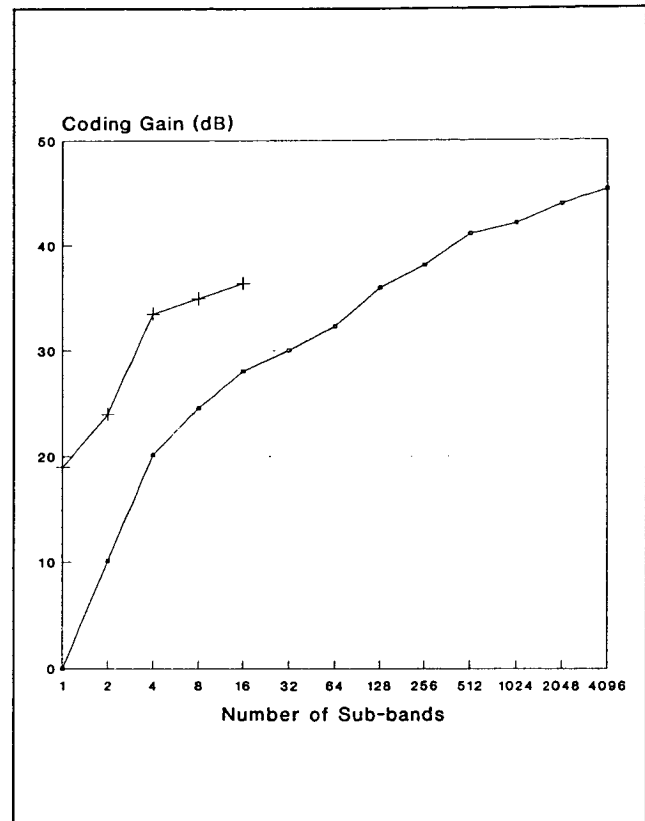


Figure 3 Composite sub-band + prediction gain versus theoretical sub-band gain

coding delay and allows the predictor to follow the short term characteristics of the signal without transmitting side information.

The success of linear predictive coding is highly dependent on the predictability or correlation of the sub-band signals. Figure 2 shows a breakdown of the sub-band prediction gains obtainable for some musical instruments. In each case the gain is reduced at higher bands, and for most instruments the gain is apparent only in the first two bands.

The coding gain of sub-band coders with and without linear prediction in each band is illustrated in Figure 3. For this trombone signal the 4-band coder with prediction is seen to give comparable gain to a 64-band coder without prediction.

1.3 Adaptive quantisation

Adaptive quantisation exploits the relatively slowly varying energy fluctuations in time which audio exhibits by continually adjusting the quantiser step size dynamically to match the signal level. This provides an almost constant and optimal signal to quantisation noise ratio throughout the quantiser operating range.

The apt-X100 coder uses a Laplacian quantiser with backward adaption which extracts the step size information from the recent history of the quantiser output. This avoids the problem of

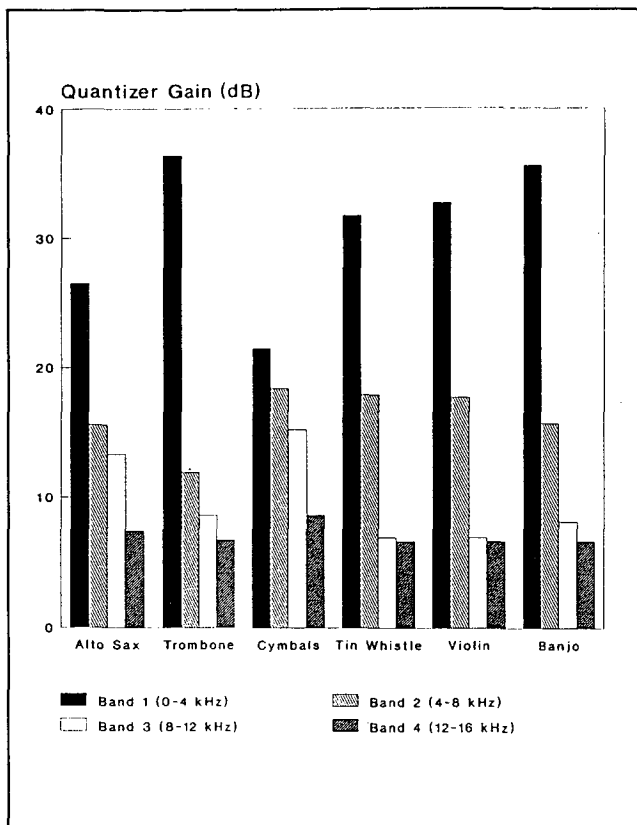


Figure 4 Variation of quantiser performance with sub-bands

estimation delay and range transmission over-heads, but has the disadvantage in that the adaption is based on reconstructed values and not the original values. This tends to limit the accuracy of the estimation and for non-stationary inputs, such as fast signal transients, an accurate estimation of the impending input signal is not possible.

The variation of quantiser performance in each sub-band for varying input signals is shown in Figure 4.

1.4 Composite algorithm (sub-band ADPCM)

The complete apt-X100 algorithm is illustrated in Figure 5. The input audio signal is filtered into four frequency bands of uniform bandwidth. Incorporated within each band is a backward-adaptive predictor and backward-adaptive Laplacian quantiser. The four code words from each sub-band are multiplexed into a single 16-bit word and transmitted to the receiver. The decoder de-multiplexes each 16-bit word and feeds each sub-band inverse quantiser with its respective code. The output from each sub-band quantiser is a difference signal which must be added to the predicted signal generated from previous samples. Finally the four reconstructed sub-band signals are inverse filtered and output as 16-bit PCM samples in the time domain.

Table I shows the objective performance of the composite algorithm averaged over four seconds of music for various monophonic signals. The equivalent PCM bits shown in the last column of Table I is a calculation of the number of bits that would be required for a linear PCM codec to obtain an equivalent quantisation noise in each sub-band.

1.5 Coding delay of composite algorithm

In apt-X100 the loop-back coding delay, that is the delay between the input of an audio signal and the output of the reconstructed signal, is 122 PCM samples, and is due to the use of linear phase,

Input	Band	Prediction Gain (dB)	Quantisation Gain (dB)	Band Gain (dB)	Equiv. PCM bits
BANJO	1	10.5	35.6	46.2	13.0
	2	2.8	15.7	18.6	13.3
	3	2.1	8.1	10.3	14.1
	4	1.1	6.6	7.7	15.3
VIOLIN	1	5.0	32.7	37.7	11.5
	2	2.8	17.7	20.5	11.7
	3	2.4	6.9	9.3	12.9
	4	1.8	6.6	8.4	14.8
TIN WHISTLE	1	21.5	31.7	53.2	12.2
	2	12.9	17.9	30.8	12.4
	3	4.2	6.9	11.1	13.4
	4	1.2	6.6	7.8	14.7
TROMBONE	1	14.3	36.4	50.7	13.2
	2	0.9	11.9	12.8	14.2
	3	0.3	8.6	8.9	14.4
	4	0.9	6.7	7.6	14.7

Table I Objective performance of 4-band ADPCM

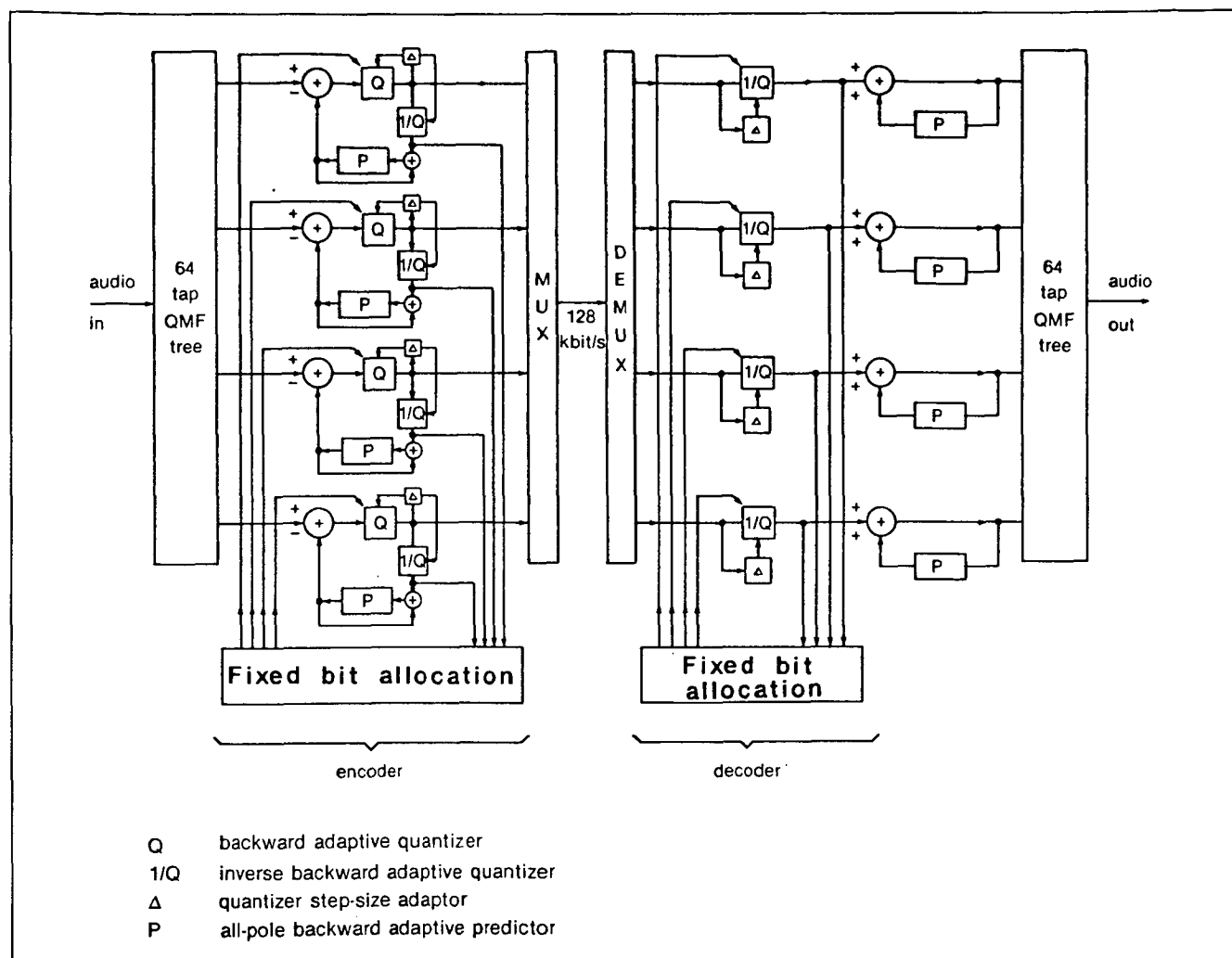


Figure 5 apt-X100 music coder

digital filter stages. Corresponding time delays for standard PCM sampling frequencies are given in Table II.

For audio storage/playback and general broadcasting, delay times are not critical. However, for applications such as audio conferencing over ISDN, or outside broadcasting where the functionality of the service is greatly enhanced if delay times are kept short, apt-X100 is ideal.

1.6 Bit error response

The apt-X100 algorithm is inherently insensitive to random bit errors, and therefore no protection of the transmitted compressed data is necessary. No audible distortion is apparent for normal programme material at a bit error rate (BER) of 1:10,000 while speech is intelligible at a BER of 1:10.

This resilience to bit errors is due to a combination of the three coding elements discussed. First of all distortions introduced by bit errors are constrained within a sub-band. In addition backward adaptive prediction and

quantisation tend to reduce the significance of the errors by spreading their affect over the trailing window of samples used for the updating. Furthermore, the magnitude of the effect of a bit error on the sub-band predictors and quantisers is proportional to the magnitude of the differential signal being decoded at that instant. Thus if the transmitted differential signal is small, which will be the case for a low level input signal or for a resonant, highly predictable input signal, then any bit error will have a small effect on either the predictor or quantiser at the decoder. The bit error response is thus well-matched to the sensitivity of human hearing, traditionally critical signals being relatively immune to errors.

The propagation of any error is halted at the decoder through the use of exponentially-tapered windows which give greater importance to the most recent samples and allow the influence of samples further back in time to 'leak' away.

1.7 Objective comparison with NICAM

The performance of the BBC's NICAM coding system is well documented and provides a useful point of reference for more recent audio coding

PCM Sampling Frequency	Time Delay Between Input Audio and Output Reconstructed Audio
32 kHz	3.8 ms
44.1 kHz	2.7 ms
48 kHz	2.5 ms

Table II Loopback coding delay for apt-X100

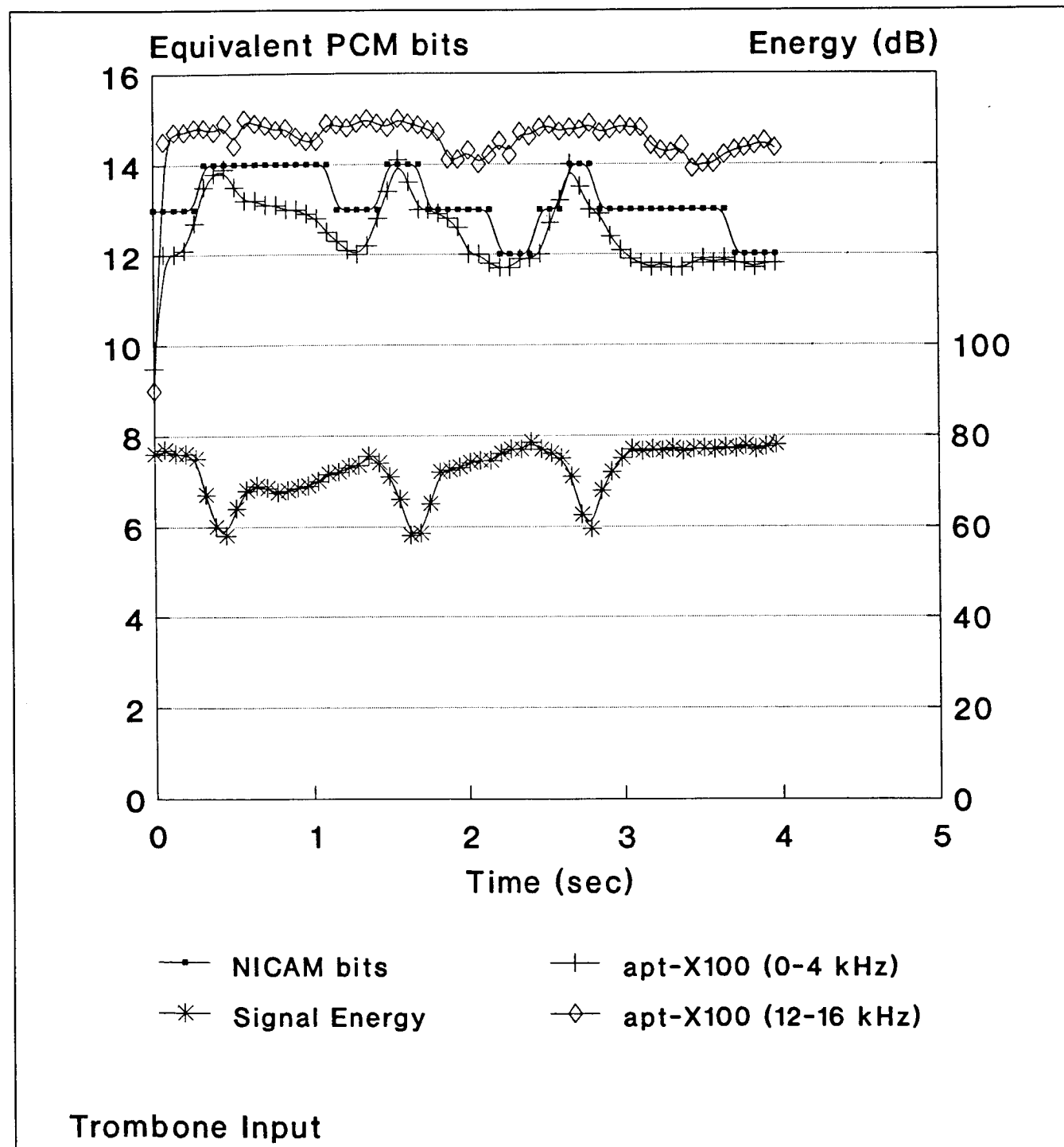


Figure 6 apt-X100 sub-band coding resolution compared to NICAM resolution

systems [1]. Figure 6 compares the objective performance of the individual lower and upper sub-bands of apt-X100 with NICAM for a trombone input signal, the energy of which is also displayed. The objective performance is again displayed as the 'equivalent PCM bits' that would be required to produce the same coding noise level. In the low energy regions of the passage the equivalent PCM resolution of the lower (0-4kHz) band is very similar to NICAM, with apt-X100 dropping slightly below NICAM during the louder sections. These results imply that the 16-bit to 4-bit coding performance of apt-X100 approaches that of the 14-bit to 10-bit coding scheme of NICAM up to 4kHz. However, the objective performance of apt-X100 equals or surpasses that of NICAM above 4kHz for this passage. This is clearly seen in the equivalent PCM bits measured in the upper bands, an example of which is illustrated in Figure 6. This shows the objective performance of the 12-16kHz band which exceeds NICAM by almost 2 bits during the loudest sections. The problem of audible noise pumping at high frequencies, to which broad-band coders such as NICAM are susceptible, is thus eliminated in the apt-X100 system. Hence the subjective performance of apt-X100 should exceed that of NICAM for this material.

2. MASKING THEORY APPLIED TO APT-X100

The outstanding audio performance of the ADPCM based apt-X100 algorithm is not surprising given the development of ADPCM over a long period of time as the standardised method of speech compression [2]. ADPCM consists of coding elements which are inherently matched to the auditory process, and the inclusion of ADPCM within a sub-band framework further improves this. Optimisation of the performance of the apt-X100 algorithm has relied on extensive listening tests, without any direct reference to auditory theory. In this section the coding performance of apt-X100 is compared to noise masking thresholds, the calculation and application of which form the basis of many transform-based audio coders.

In digital audio coding an auditory masking theory is usually invoked in an attempt to determine the irrelevant parts of an audio signal. Irrelevant parts of the signal are those which are deemed inaudible to the human ear, due to their being masked by higher level signals at different and usually lower frequencies. To determine irrelevancy involves characterising and modelling human hearing, and applying this model, or an approximation of it, to the audio signal. The final result is a frequency-dependent masking threshold, which gives an indication of the level of noise at any frequency that may be injected into the original signal without audible effect. Any part of

the original signal that falls below this masking threshold may be ignored, and in addition those parts that remain can be quantised to the level of noise indicated by the threshold at that particular frequency. Thus the noise masking threshold serves to remove irrelevant spectral parts of the signal and to reduce the level of coding accuracy needed for the remainder.

ADPCM on the other hand is a procedure which aims primarily to determine and remove redundancy from an audio signal, that is, to remove those predictable parts of the signal prior to coding which may be reconstituted again, so as to accurately reconstruct the original signal [3]. As such ADPCM does not explicitly take into account the auditory properties of the human ear, redundancy being a purely objective measurement of any signal. Quantisation noise introduced to the signal during coding is added without any knowledge of the allowable noise masking thresholds. The apt-X100 algorithm, by using four frequency bands, simply constrains quantisation noise to within these bands. It is of interest therefore to compare the noise injected by apt-X100 with the noise masking thresholds calculated from a psycho-acoustical model of hearing.

The model used to calculate the noise masking thresholds is based on Johnston [4] and involves transforming the signal into the frequency domain, partitioning the power spectrum into 'critical bands', calculating masking thresholds in each band and finally spreading the thresholds of each band across all bands. This procedure for determining the noise masking thresholds does not take into account temporal masking effects.

Figure 7 is a spectrum of the glockenspiel (EBU SQUAM CD 422-204-2 track 35) and its associated noise masking threshold, and Figure 8 illustrates the error spectrum generated from apt-X100 for this signal. The frequency scale is an approximation to the bark scale, each critical band being of uniform width. The most important feature is that the error signal is generally below the masking threshold at all frequencies, and is typically flat across each of the four sub-bands. At the resonant frequencies the noise is considerably below the allowable threshold. This is intuitively reasonable given that resonant signals tend to be predictable and hence exhibit redundancy which may be removed from the signal. What is perhaps surprising is that the error signal is below the masking threshold even in the noisy regions of the spectrum, where the predictability and redundancy of the signal is small. It is also noteworthy that while the use of the masking threshold will tend to remove large portions of the spectrum, particularly at the high frequencies, apt-X100 will always code these signal components, irrespective of their level.

In this particular spectrum therefore, apt-X100,

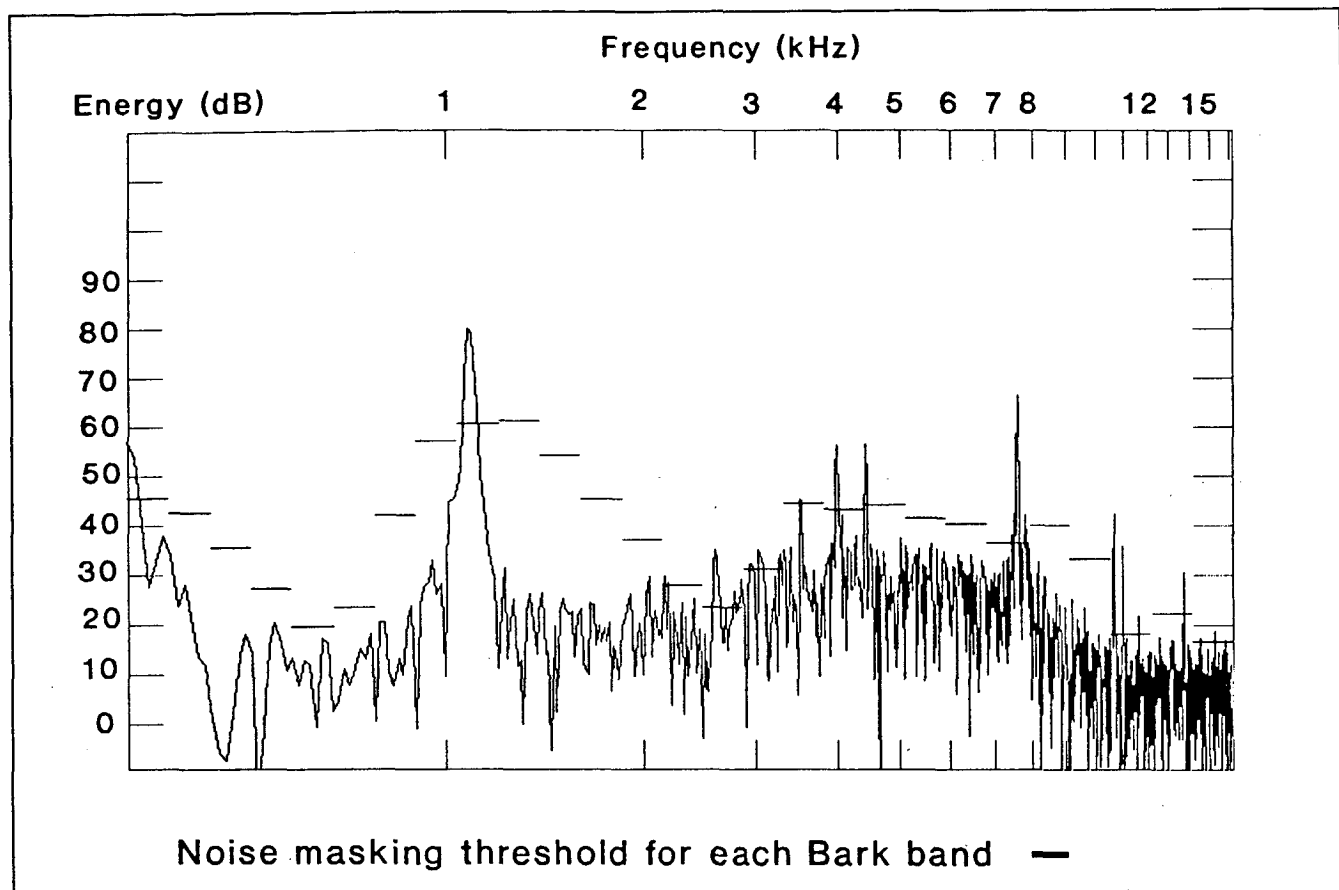


Figure 7 Spectrum and masking thresholds of glockenspiel

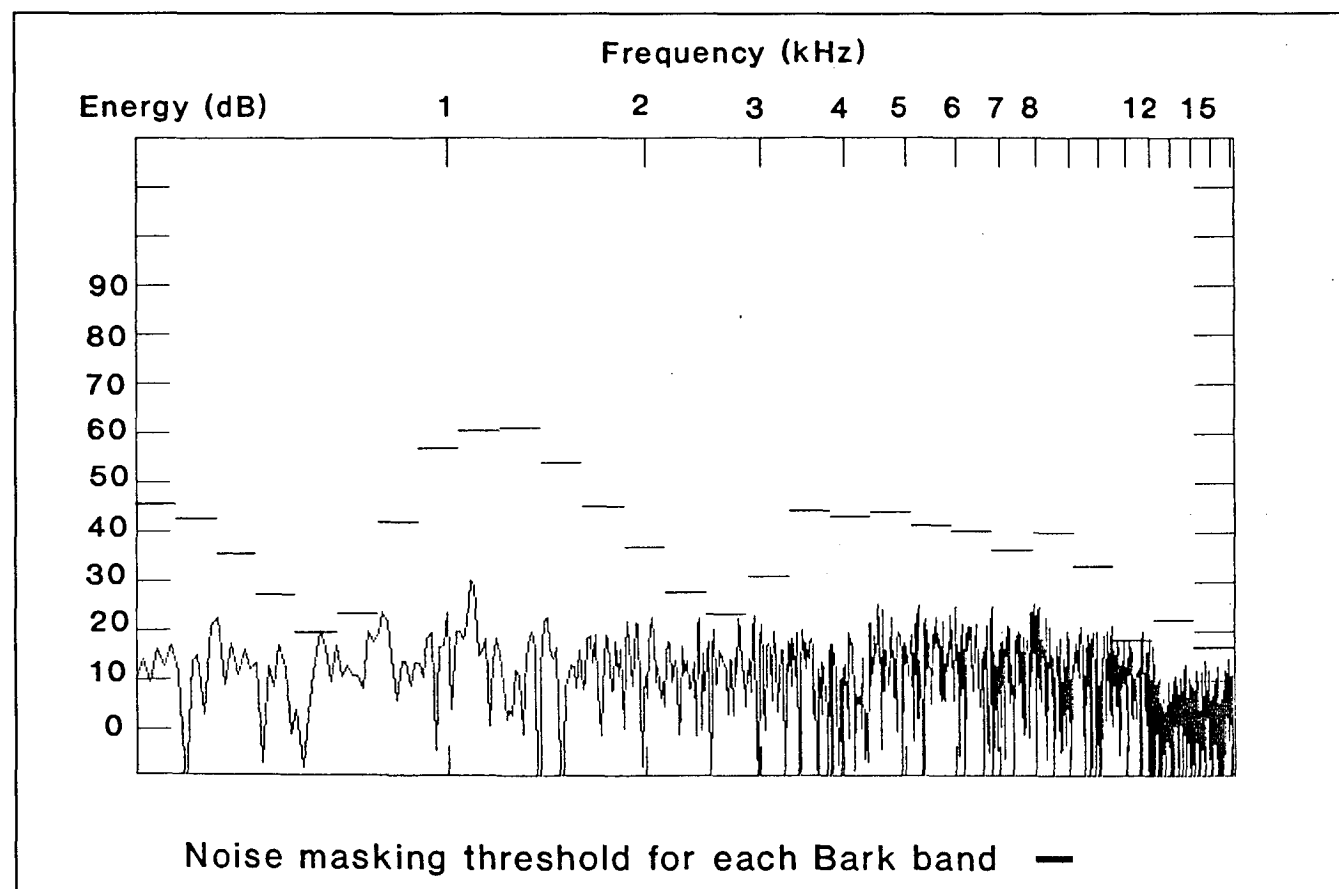


Figure 8 apt-X100 error spectrum of glockenspiel

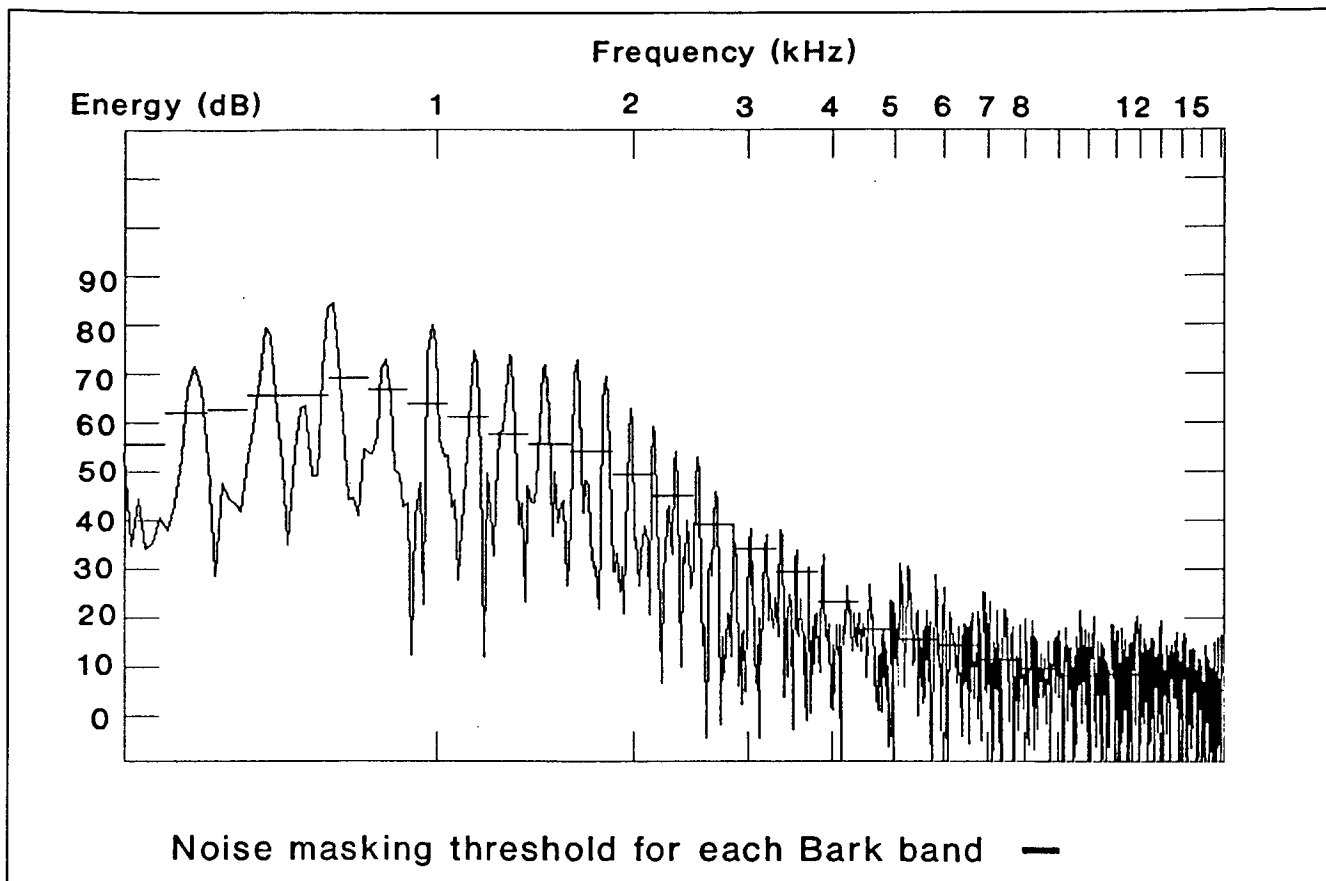


Figure 9 Spectrum and masking thresholds of trombone

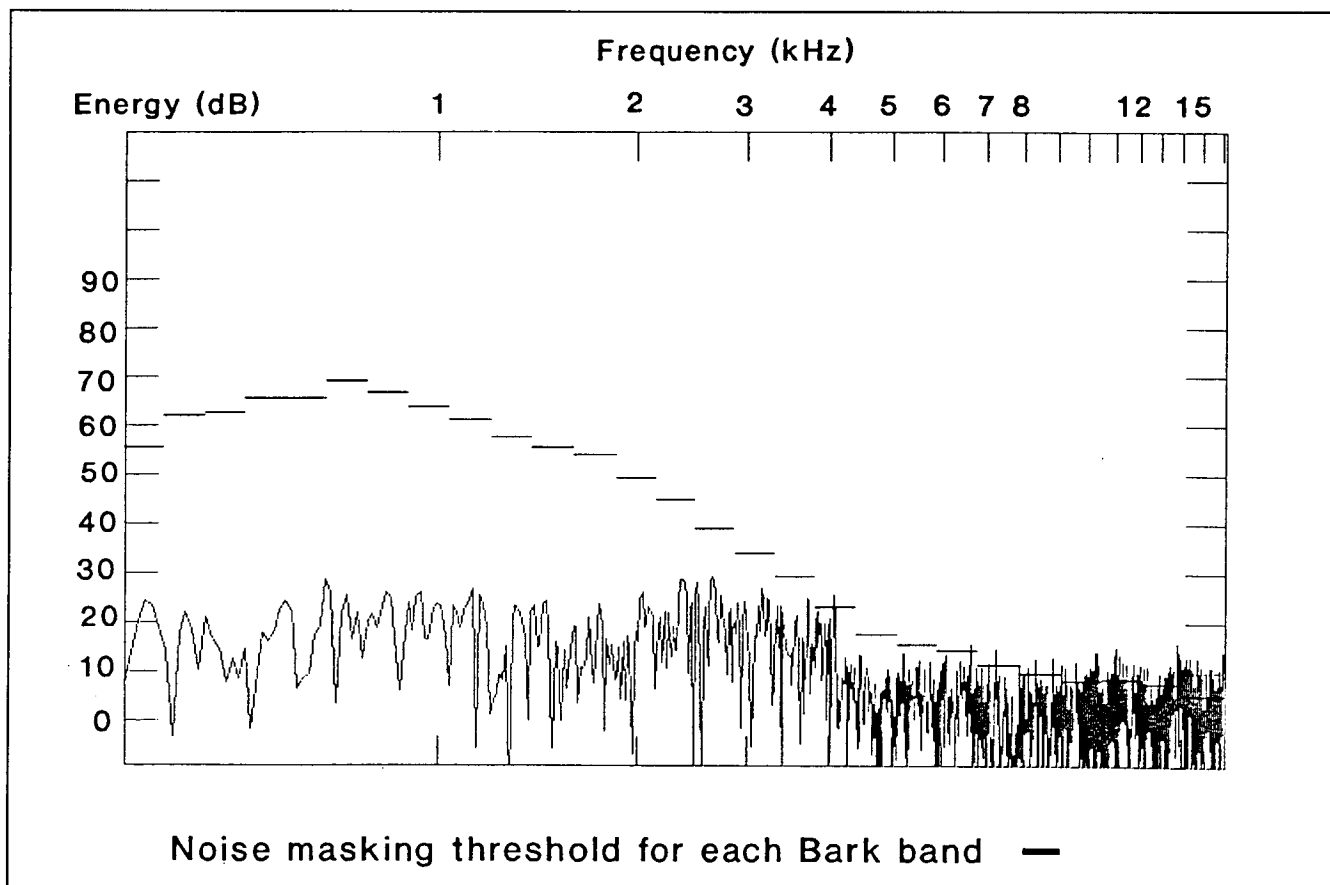


Figure 10 apt-X100 error spectrum of trombone

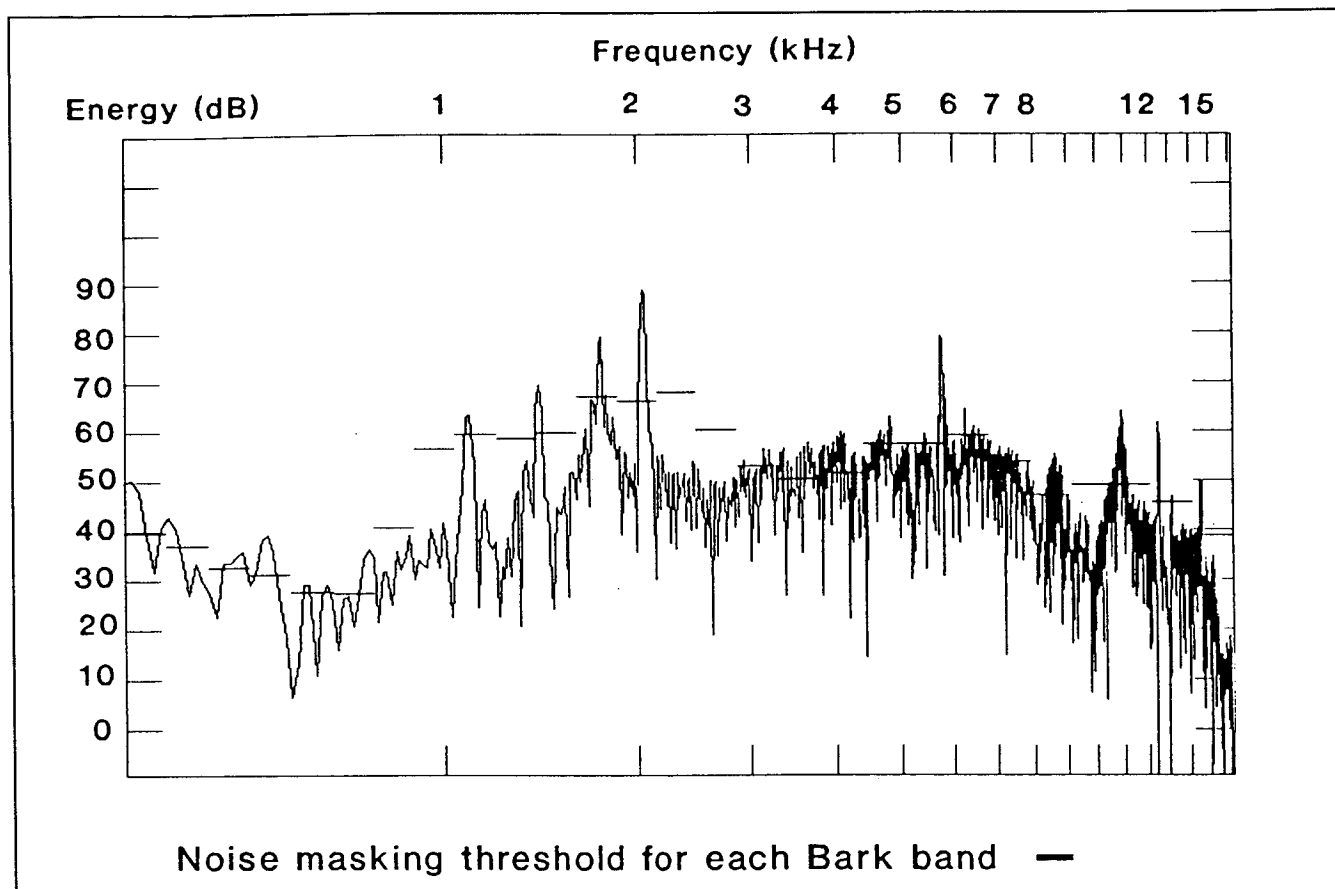


Figure 11 Spectrum of glockenspiel (transient)

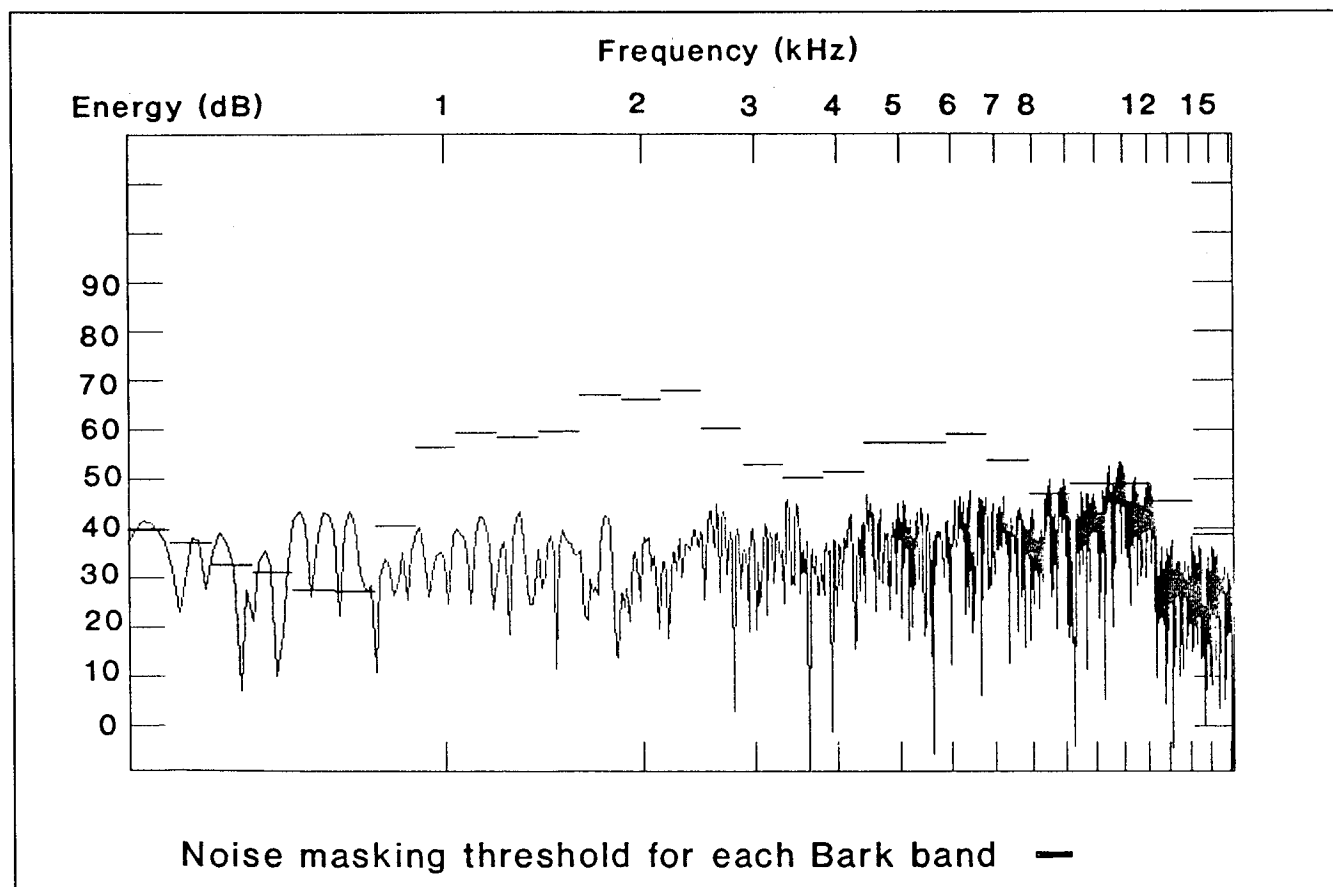


Figure 12 apt-X100 error spectrum of glockenspiel (transient)

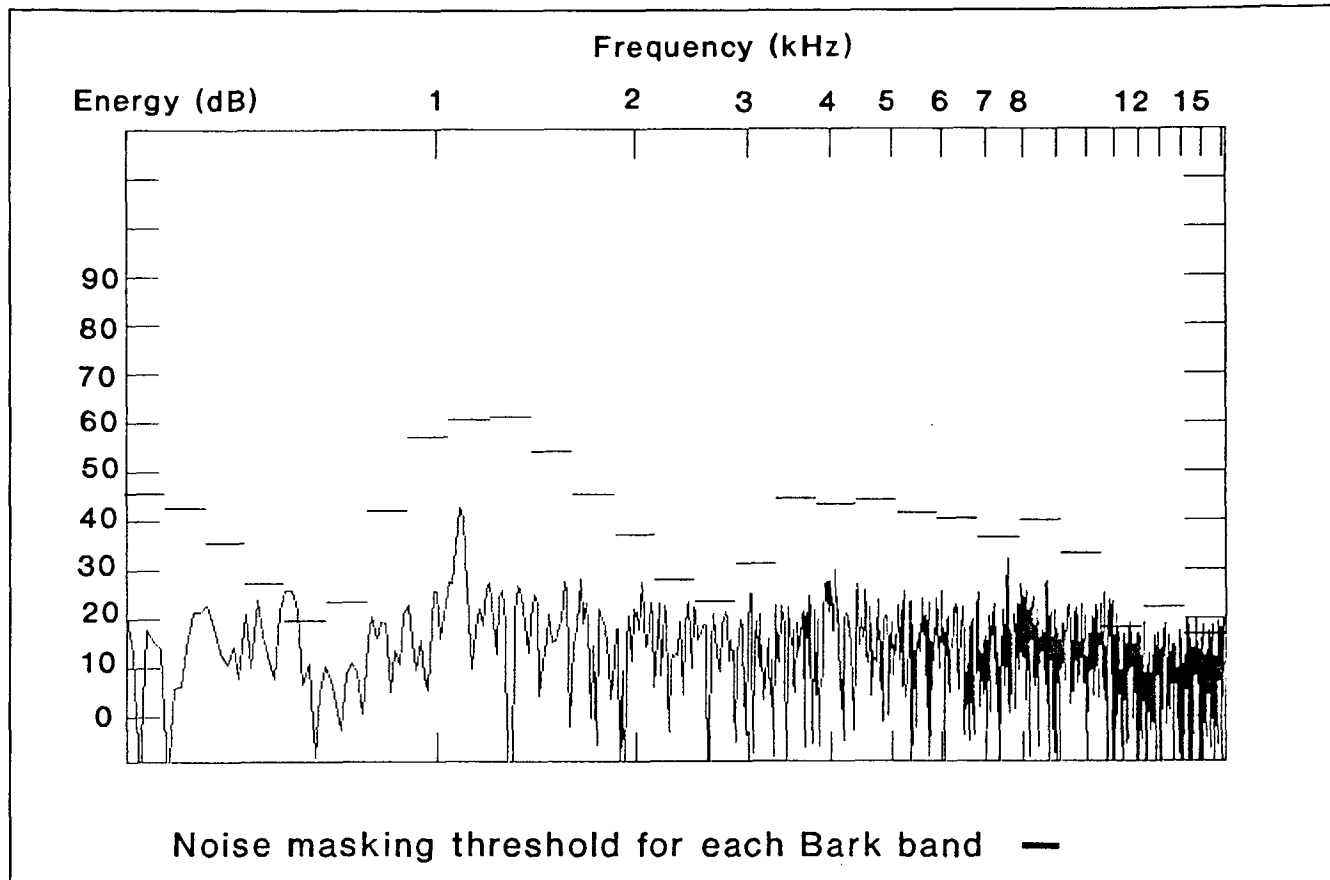


Figure 13 apt-X100 error spectrum of glockenspiel (5 passes)

whilst not fully exploiting the noise masking threshold in the resonant regions, nevertheless does not exceed it elsewhere. This is also seen in Figures 9 and 10 which show the spectrum, error spectrum and noise thresholds of a trombone (EBU SQUAM CD 422-204-2 track 22).

For a transient signal (Figure 11), also of a glockenspiel, the error signal (Figure 12) is considerably increased and tends to rise above the noise threshold typically in the 8-12kHz sub-band. This general increase in noise is also reasonable given the unpredictability of transient signals. However, the audible effect of this noise breakthrough is wholly alleviated by temporal masking, a psycho-acoustic phenomena which is explicitly accounted for within the apt-X100 algorithm by the fast-attack/slow-decay characteristic of the adaptive step-size of the quantiser.

Given that these examples show that apt-X100 does take account of the characteristics of human hearing without any explicit calculation of the noise masking threshold, what advantage is gained from not fully exploiting the threshold? Some applications require an ability to post-process the compressed data in the digital PCM domain, and subsequently re-code the data. If all irrelevancy is removed from an audio signal in the first coding pass, then by definition tandem coding will cause the cumulative noise to exceed

the masking threshold and become audible. Coders which rely on a calculation of the noise masking threshold for their operation counter this problem by a more conservative use of the masking theory, for example, by reducing the level of noise introduced into the critical bands containing resonant structure [5]. This allows for a limited degree of tandem coding albeit with a slight audible degradation between passes and at the expense of less compression. The apt-X100 algorithm, by not fully exploiting the noise masking threshold especially in resonant regions, is much more tolerant of tandem coding. This is demonstrated in Figure 13 which shows the error spectrum of a glockenspiel signal (Figure 7) after five digital passes through the apt-X100 algorithm. Each apt-X100 coding passage accumulates noise but at a decreasing rate leading to a gradual degradation in audio quality with the number of coding passes (Figure 14).

In general apt-X100 is well-matched to the overall characteristics of human hearing in that critically perceived signals such as pure tones are highly redundant and hence accurately coded, whilst noisy signals which are not predictable and therefore poorly coded are not critically perceived. Comparison with noise thresholds calculated from a masking theory show that apt-X100 induced noise is consistently below audible levels. Moreover the audio bandwidth of apt-X100 is simply one half of the sampling rate, irrespective

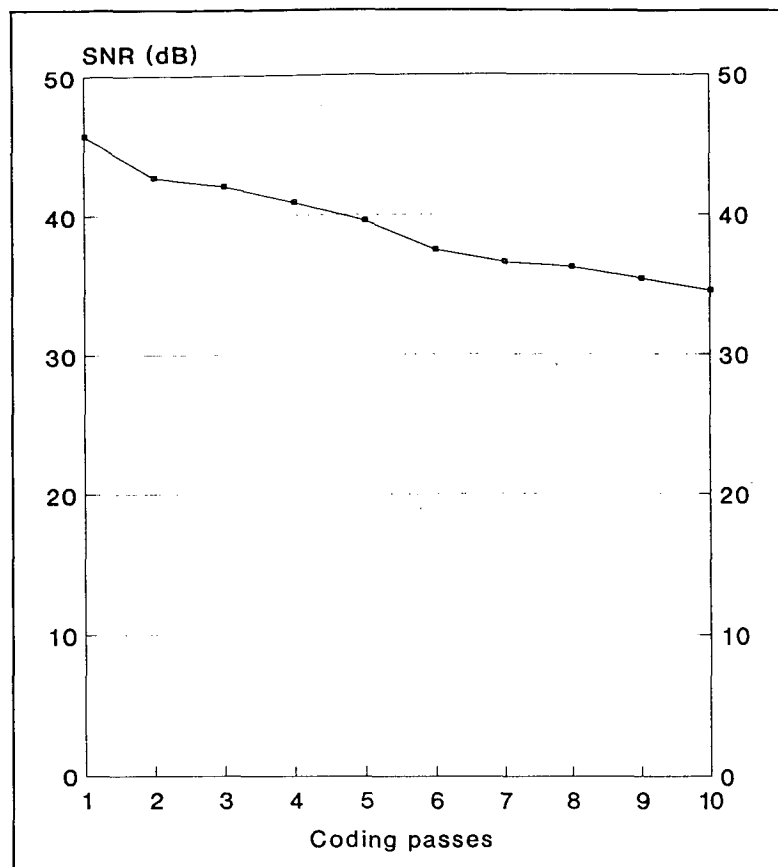


Figure 14 Variation of SNR with coding passes (apt-X100)

of its absolute value. This is of relevance in applications which demand operation at different sampling rates.

3. HARDWARE IMPLEMENTATION

3.1 Chip level

The apt-X100 coding system is currently available on a single AT&T DSP16 digital signal processor with separate devices being required for the encoding and decoding stages, the APTX100E and APTX100D respectively. With 512 X 16 words RAM and 2048 X 16 words of program ROM included in the DSP16, it has been possible to store and run either encoder or decoder algorithms on masked ROM versions of this device without the need for external memory. As a result, the hardware design around the apt-X100 processors is straightforward, requiring normally only the PCM convertor and some basic timing logic to complete an audio data compression system.

Recently dual channel apt-X100 encoding and decoding algorithms have been implemented on a single ROM-masked AT&T DSP16A digital signal processing chip. In stereo applications, the use of a single DSP16A device instead of two monophonic DSP16 chips, is particularly economic in high volume projects.

Apart from the main sub-band ADPCM processing blocks, all apt-X100 devices also include many additional software features to enhance their functionality. These include automatic synchronisation, auxiliary data transmission and PCM level detection.

3.2 Automatic synchronisation (AutoSync)

Automatic synchronisation is a facility which enables the apt-X100 compressed data stream to be decoded without prior knowledge of the 16-bit compressed word boundaries. As a result the use of AutoSync enables the compressed data to be handled at both transmitter and receiver using only bit timing, no word clock being required. No bandwidth overheads are incurred through the use of any of the AutoSync facilities.

Synchronism is obtained by inserting into the compressed audio data stream a unique sync word. This is searched for at the decoder and once found establishes the compressed word boundaries for certain multiplexed formats. Insertion occurs once every 128 16-bit compressed words (frame length = 2048 compressed bits), and to establish synchronisation three consecutive sync words must be found by the decoder. This allows for synchronisation under adverse channel error conditions as summarised in Table III.

Loss of sync lock is flagged at the decoder, and the PCM output muted if sync words are not found in three consecutive frames. Synchronisation from processor power-up or bit errors is therefore achieved within two to three frame periods, while re-synchronisation is achieved within four to six frames from the instant of clock slip. The time taken to (re)synchronise is thus entirely dependent on the sampling rate.

In addition to synchronising a single channel, AutoSync also enables two, four or eight channels to be multiplexed/de-multiplexed, again with no bandwidth overheads. Only one encoder processor is used to send a sync word, whilst each decoder processor uses this sync word to de-multiplex any one channel out of the original two, four or eight channels.

AutoSync combines powerfully with the 16-bit word format of the compressed audio data stream, which is essentially identical to standard 16-bit audio PCM. For example, the AutoSync facility enables the direct replacement of a single 16-bit, 32kHz sampling rate, 1024kbit/s stereo audio link with four apt-X100 compressed stereo channels with no change in timing circuitry.

	Bit Error Rate (BER)		
	1:10	1:100	1:1000
Occurance (failures /min)	14	0.2	0
Maximum duration (frames)	5	3	0

Table III AutoSync lock failure under random bit errors

3.3 Auxiliary data

Auxiliary data may be transmitted between the encoder and decoder chips by replacement of the least significant bit of each 16-bit compressed data words. This bit stealing can be dynamically switched on or off according to line activity through the use of the AutoSync which has the facility of allowing the encoder to remotely enable the auxiliary data mode of the decoder. The data bit rate for transmission of auxiliary data per mono channel is 1/4 of the sampling rate, allowing standard 9600 baud transmission within a 32kHz stereo audio link, and up to 24kbit/s within a 48kHz stereo link. This opens up the possibility of a talk-back facility during normal audio transmission between transceivers, toll quality speech (3400Hz bandwidth) being commonly realised at 16kbit/s.

3.4 PCM threshold level detection

This facility compares the absolute value of the input PCM signal averaged over four samples with one of 64 possible threshold levels held in a table within the encoding processor. If the PCM level is above the threshold level a THRES flag is set high. Allowed threshold levels range from -96dB in 1.5dB steps and are set by applying the appropriate table address to the encoder device.

3.5 Stereo coding/decoding boards

Stereo encoder (SCS100) and decoder (SDS100) boards have been developed which allow easy access to every facility offered in the apt-X100 algorithm, and are primarily intended for OEM's. Use of 64 times oversampling dual-channel A/D and D/A converters allow sampling frequencies to vary continuously between 8kHz and 48kHz with no additional filtering. The boards are designed to operate in either stereo or mono.

The stereo encoder board accepts either balanced analogue audio or digital audio (AES/EBU or S/PDIF formats) inputs, with a further option of receiving direct TTL data from a remote ADC. The compressed output data is in 16-bit word format, each word representing four PCM samples. Data output and all output timing signals are RS422 (balanced). The sampling

frequency of the ADC may be set either internally by an on-board oscillator, (active clock operation) or externally by an input clock, (passive clock operation). Auxiliary data is supported as TTL input from a synchronous serial port.

The stereo decoder board operates passively only, the minimum requirement for operation being bit clock timing, to establish the sampling rate, plus compressed data (AutoSync enabled). The compressed input data stream may comprise one, two, four or eight multiplexed channels, any two of which can be selected for decoding. Audio is output either in analogue from the dual-channel 16-bit DAC, or in digital format (AES/EBU or S/PDIF), or from a remote DAC (direct TTL output of PCM). All timing and compressed data signals are RS422 (balanced). With AutoSync disabled, a word clock must be supplied to define the compressed code word boundaries. Auxiliary data is output as TTL.

3.6 PC expansion card (ACE100)

The ACE100 PC expansion cards have been recently introduced and are primarily intended as a low cost platform on which OEM's can develop their own application specific software. The card essentially allows two channel record and play-back of apt-X100 compressed audio or 16-bit PCM audio at user-selected sampling frequencies directly to and from a PC hard or floppy drive. A noteworthy feature of the ACE100 is the inclusion of a user-programmable DSP which has direct access to all the main data busses. As a result, both pre- and post-processing of the audio data is facilitated. Up to four cards may be used simultaneously in any single PC, allowing the possibility of an 8-track digital audio recorder.

3.7 Duplex stereo transceiver (DSM100)

For broadcasters a complete stereo encoding/decoding unit has been developed, the DSM100. This unit utilises the stereo codec boards described above. Audio I/O is either balanced analogue or digital (AES/EBU or S/PDIF), with an unbalanced microphone input and headphone output also provided. Compressed data I/O is via a RS449 interface, the data rate being determined externally by the clock frequency of the

Mode (encode or decode)			Processor Version (AT&T DSP16A cycle time)		
			55 nsec	33 nsec	25 nes
32	kHz	mono	yes	yes	yes
44.1	kHz	mono	no	yes	yes
48	kHz	mono	no	yes	yes
32	kHz	stereo	no	tba(*)	yes
44.1	kHz	stereo	no	no	tba(*)
48	kHz	stereo	no	no	tba(*)

Table IV Modes of stereo encode/decode processor

transmission line, or internally by an on-board crystal oscillator clock on the encoder card. A full duplex RS232 auxiliary data interface is also provided, allowing up to 19200 baud transmission between DSM100 transceivers (sampling frequency 48kHz). Visual indicators of the input and output stereo signal levels, current mode of operation, status and errors are provided, relevant signals being fed to an external alarm port.

4. STEREO ALGORITHM

A very recent development is the implementation of a stereo encoder and stereo decoder algorithm on a single masked version of the AT&T DSP16A signal processor. This processor has 2k x 16 RAM and 12k X 16 ROM which is sufficient internal memory for storing two monophonic encoder algorithms and two monophonic decoder algorithms, and for running the chip either as an encoder or decoder in stereo or mono. The range of operating modes of this signal processor are outlined in Table IV.

Furthermore a more efficient implementation of the algorithm which is under development will enable the 33ns version to encode or decode 32kHz stereo and the 25ns version to process 44.1 and 48kHz stereo audio.

The stereo encode/decode chip can dynamically switch its mode of operation from encoding to decoding. In applications such as audio storage/playback this ability to swap operating modes makes the chip very cost effective.

5. CONCLUSION

apt-X100 is a very high quality 16 to 4-bit audio compression system designed specifically to reduce by a factor of four the amount of data associated with PCM digital audio without affecting the sound quality. Use of the system allows an enormous saving to be made in those

applications which normally transmit or store PCM digital audio signals.

The apt-X100 process is based on sub-band ADPCM techniques which utilise both signal redundancy and auditory masking to achieve transparent coding at 4-bits per sample. This has been demonstrated by comparing coding noise introduced by apt-X100 with noise masking thresholds calculated from a relevant auditory model. Moreover, because the noise levels are generally well below the calculated values, apt-X100 has been shown to be tolerant of tandem coding.

A summary of the operational features of the apt-X100 system must include:

- subjectively and objectively optimised
- operation at any sampling frequency to 48kHz
- inherent tolerance of tandem coding and post-processing
- inherent immunity to bit errors better than 1:10,000
- very low coding delay (< 4ms at fs = 32kHz)
- linear phase response and low ripple across bandpass
- low hardware complexity and cost

apt-X100 is currently available in four formats to allow entry of the technology at any design level.

1. The lowest level is pre-programmed encoding and decoding chips (Figure 15), which allow OEMs to implement and specify apt-X100 without the need to formally license the technology. Each chip incorporates features such as automatic data synchronisation and de-multiplexing schemes

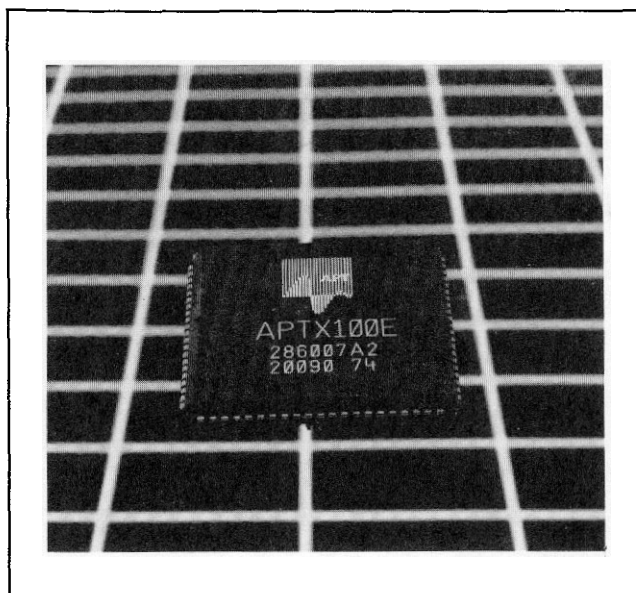


Figure 15 apt-X100 encoder

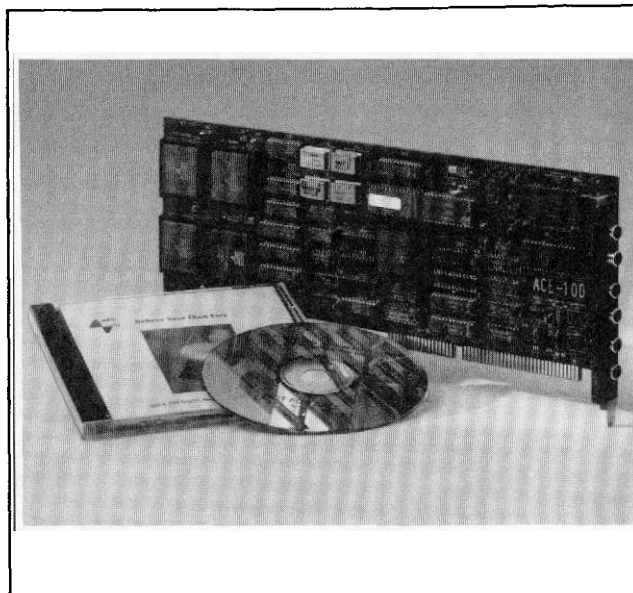


Figure 17 ACE100 PC expansion card

to facilitate zero overhead decoding of data in asynchronous environments. Hence for many applications only limited additional timing circuitry is necessary

2. The next level is based on stereo encoding and decoding circuit boards (Figure 16) which are completely self-sufficient, having on-board audio conversion, clock regeneration and data I/O circuitry. Both boards have been designed with OEM's in mind being capable of operating in a broad range of applications
3. For ease of use within the PC environment a PC-based expansion card (ACE100 Figure 17) has been designed which allows for the real-time recording and play-back of compressed stereo audio directly onto either floppy or hard drives. Pre- and

post-processing is supported through the use of a user-programmable DSP which controls all data movements between the ADC, DAC, apt-X100 devices and main memory

4. The highest level is a fully integrated digital audio transceiver unit (Figure 18). In addition to the facilities offered by the stereo encode/decode boards this unit offers RS449 compressed data I/O, RS232 auxiliary data, and a choice of balanced analogue or digital (AES/EBU S/PDIF) I/O

apt-X100 is today a widely used audio coding system, being specified in many products in the broadcasting and professional audio markets. This proliferation is primarily due to its singular high quality, wide applicability and low hardware complexity.

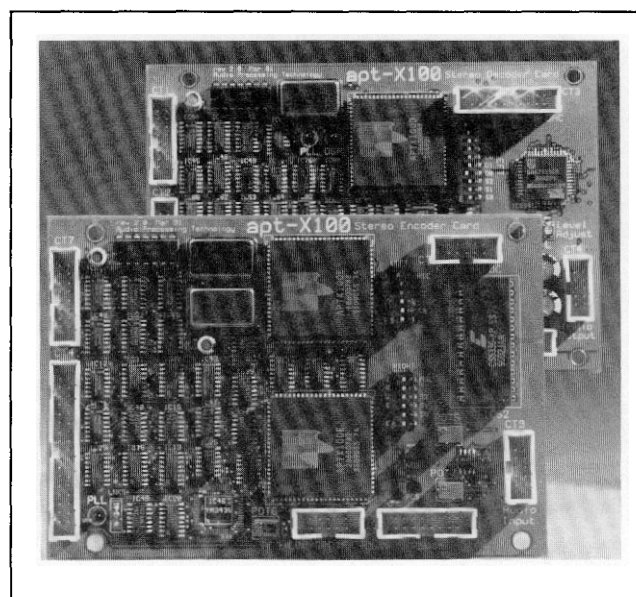


Figure 16 Stereo encoder and decoder cards

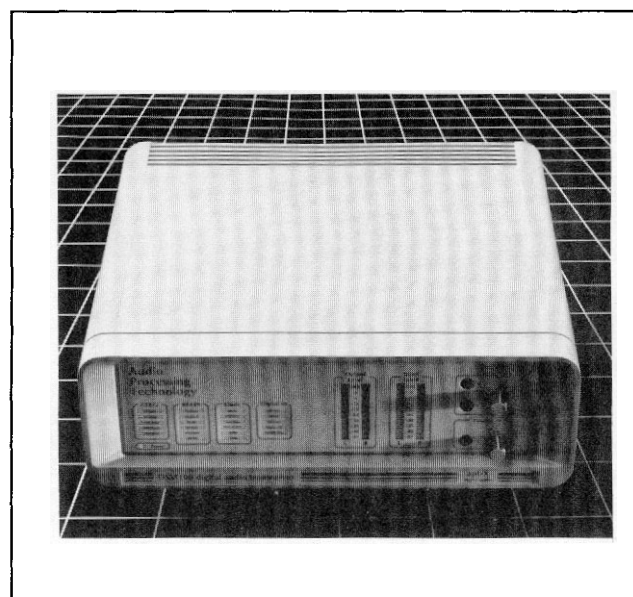


Figure 18 DSM100 digital audio transceiver

REFERENCES

- [1] Soumange J., Mabillet P.: "A comparative study of the proposed high quality coding schemes for digital music" Proc. ICASSP, April 1986
- [2] CCITT, Study Group XVIII, Report R26(c)n Recommendation G.72x, August 1986
- [3] Smyth S.M.F.: "Digital audio compression - a practical solution" Proc. NAB Engineering Conference 1989, pp 234-241
- [4] Johnston J.D.: "Estimation of perceptual entropy using noise masking criteria" ICASSP-88 New York pp. 2524-2527, April 1988
- [5] Thiele G., Stoll G., Link M.: "Low bit-rate coding of high quality audio signals. An introduction to MASCAM" EBU Review - Technical No.230, Aug 1988