EXTENDED-BAND AUDIO CODING EXPLOITING SPECTRAL-DOMAIN SAMPLING-RATE CONVERSION WITH EMBEDDED ULTRASONIC DATA

MOJ Hawksford Emeritus Professor, University of Essex, Colchester, CO4 3SQ
mjh@essex.ac.uk

ABSTRACT

A method is described that embeds extended-band data within the least significant bits of LPCM data while retaining compatible replay to greater than CD quality without decoding. Pseudo-randomised embedded data occupies the lower data bits by signal dependent means enabling dynamic exchange between extended-band and audio-band signal-to-noise ratio. Ultimately this allows the dynamic range of the audio-band signal to closely match the full bit depth of the LPCM data. Also, output data is structured so if truncated to 16 bit, an uncompromised CD-quality audio file results. Precision down-sample rate conversion is achieved using a spectral-domain method that naturally outputs complex samples and facilitates band splitting prior to ultrasonic coding. The signal processing methodology is described with experimental software able to process source data in either LPCM or DSD formats. The audio-band information is essentially transparent within the bounds of bit depth, with psycho-acoustic trade-offs targeted primarily at ultrasonic data because of its naturally low audibility.

1 INTRODUCTION

There has been much debate and conjecture regarding the merits of high-resolution audio especially where audio signals are allowed to extend into the ultrasonic audio band. At one level bandwidth extension appears illogical as human hearing rarely can detect signals exceeding 22 kHz. However, there are a number of factors that need to be considered as, for example, one school supports human perception of transient signals while another focuses on signal processing artefacts. It is a complicated and somewhat controversial issue although by applying *Occam's Razor* it maybe a case where the simplest explanation is potentially the most rational. Consequently in this paper the discussion is limited to the supposition that the less a signal is modified the more faithful it is to its origin.

The principal topic of this paper is a coding method for high-resolution audio signals where a wide-bandwidth audio signal is initially down-sampled [1] but by using a spectral-domain method [2] to form complex samples at twice the Nyquist sample rate (e.g. 88.2 kHz). This signal is then sub-divided into two equal bands 0 to 22.05 kHz and 22.05 kHz to 44.1 kHz to facilitate coding by using a method previously introduced at the *Audio Engineering* IEE Colloquium in London, 1st May 1995 [3] and illustrated in Figure 1. For purposes of definition the lower band is designated as the *audio base band* (ABB) while the upper band is the *ultrasonic audio band* (UAB).

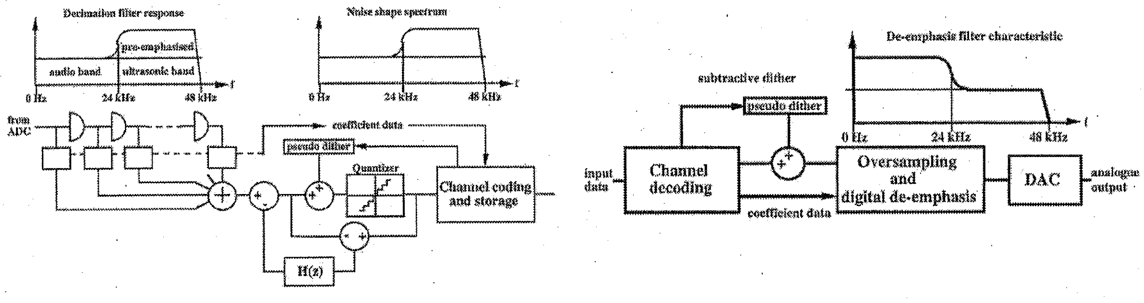


Figure 1 Outline high-resolution audio coding scheme, *Monday 1st May 1995.*

This concept [3,4] is developed here into a working algorithm where the primary objective is to retain maximum signal integrity within the *ABB*. This requires precision down-sample rate conversion (DSRC) with extremely low frequency response ripple and phase-response error. The ABB must retain a minimum dynamic range dictated by 16-bit linear pulse-code modulation (LPCM). However the proposed system allows additional dynamic range commensurate with 24-bit LPCM since the lower 8 bits employed to encode the UAB are dynamically assigned as a function of ABB signal level. The rationale is that when the ABB signal level is low it is more appropriate to encode low-level ABB signals accurately that are above the threshold of hearing rather than UAB signals that are inaudible. Also for higher signal levels low-level noise tends to be masked so 16-bit precision is sufficient. Additional data savings are also possible by mixing UAB signals to mono especially for lower level signals. A consequence of this strategy is that if a 24-bit, 44.1 kHz encoded signal is truncated to 16-bit resolution, this may be used directly without any form of decoding. In addition the full resolution encoded signal can be played directly via a conventional 24-bit digital-to-analogue converter (DAC) as up to 8 of the least significant bits are randomised so contribute only low-level noise that is always below that of CD. Also as the overall signal level falls some of the lower 8 bits are dynamically transferred to the ABB signal ultimately to achieve full 24-bit resolution at low level, again without additional decoding, since randomization is applied only to the actual bits assigned for UAB encoding. However, when fully decoded all these advantages are retained while being augmented by bandwidth extension and a doubling in sample rate.

The coding scheme is divided into two core process regimes namely, the DSRC algorithm and the UAB coding scheme. DSRC is implemented by a recently published algorithm [2] that exploits the spectral-domain matching technique incorporating matrix arithmetic and a method of complex sampling. This algorithm can handle both LPCM and DSD audio and naturally outputs complex LPCM audio data with low time dispersion. Critically, it is capable of extremely low ABB distortion with all spectral and non-linear artefacts well below 150 dB depending upon arithmetic precision. Because spectral domain DSRC is integral to the coding algorithm we commence with a brief review of this algorithm although greater background together with a discussion on problems arising in conventional DSRC methods are presented in the supporting publication [2].

2 SUMMARY OF SPECTRAL-DOMAIN DSRC

Spectral-domain DSRC eliminates the need for a conventional convolutional low-pass filter with its extended time dispersion and because aliasing distortion is virtually non-existent as a result of precise ABB spectral-domain matching. Spectral matching is performed over finite input and output data sequences designated *bitframes* (see Figure 2) where it is shown to yield accurate amplitude and phase response matching with modest finite time dispersion defined by the output bitframe length. Advantageously, having to make a choice of linear phase, minimum phase or an apodizing filter [5] based upon controversial subjective criteria does not arise.

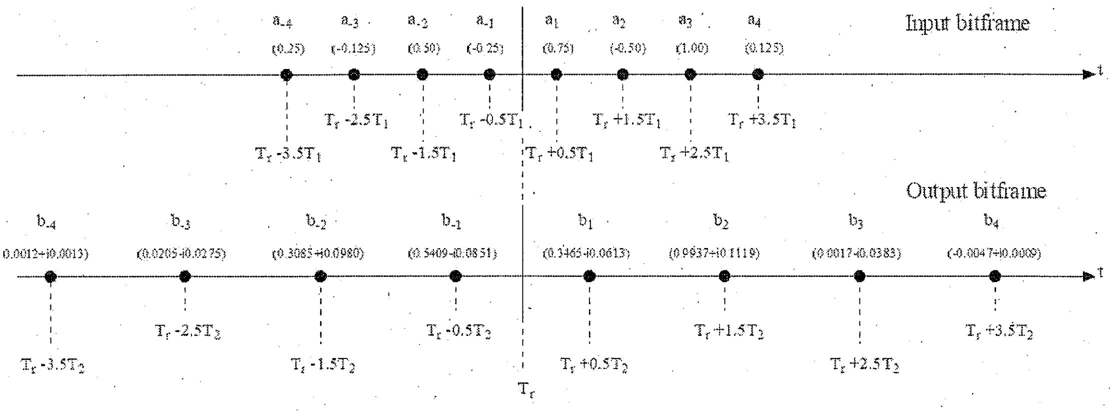


Figure 2 Input and output bitframes *r* centred on time T_r for DSRC.

A spectral-domain matching example (including a numerical worked example) for input and output bitframes each of length 8 is illustrated in Figure 2 where the source sampling period is T_1, the down-converted sequence period is T_2 and the down conversion ratio is 2. The source samples $\{a_{-p}, a_p\}_{p=1}^{M/2}$ are divided into sequential *input bitframes* of length M (where here M = 8) centred on times T_r while the down-converted samples $\{b_{-q}, b_q\}_{q=1}^{N/2}$ are divided into corresponding *output bitframes* of N samples (here N = M). The aim is to calculate the output bitframe such that over the Nyquist band of 0 to $0.5/T_2$ Hz its complex spectrum is matched to the corresponding input bitframe complex spectrum. As a result of spectral matching each output bitframe consists of a set of complex numbers meaning complex output samples. Hence, for input and output bitframes r, the corresponding spectra $\overline{A_r}$ and $\overline{B_r}$ are calculated respectively as,

$$\overline{A_r} = e^{-2\pi f T_r} \sum_{p=1}^{0.5M} \left\{ a_{-p} e^{2\pi f(p-0.5)T_1} + a_p e^{-2\pi f(p-0.5)T_1} \right\} \qquad \text{... 2-1}$$

$$\overline{B_r} = e^{-2\pi f T_r} \sum_{q=1}^{0.5N} \left\{ b_{-q} e^{2\pi f(q-0.5)T_2} + b_q e^{-2\pi f(q-0.5)T_2} \right\} \qquad \text{... 2-2}$$

The design target is to determine the set of output samples $\{b_{-q}, b_q\}_{q=1}^{N/2}$ such that,

$$\sum_{q=1}^{0.5N} \left\{ b_{-q} e^{2\pi f(q-0.5)T_2} + b_q e^{-2\pi f(q-0.5)T_2} \right\} \Bigg|_{f=0\to 0.5/T_2} \approx \sum_{p=1}^{0.5M} \left\{ a_{-p} e^{2\pi f(p-0.5)T_1} + a_p e^{-2\pi f(p-0.5)T_1} \right\} \Bigg|_{f=0\to 0.5/T_2} \qquad \text{... 2-3}$$

Spectral matching is performed by solving a set of simultaneous equations linking sample coefficients $\{b_{-q}, b_q\}_{q=1}^{N/2}$ to $\{a_{-p}, a_p\}_{p=1}^{M/2}$ when generated at discrete frequencies spanning the Nyquist band of the output sequence. Hence defining matrices $[E]$ and $[F]$ relating to the complex exponential terms of input and output bitframes respectively, Equation 2-3 may be expressed in matrix form as,

$$[F][b] = [E][a] \qquad \text{... 2-4}$$

Solving using the pseudo-inverse matrix function so that $[b]$ is optimized in the least mean-square error sense,

$$[b] = pinv([F])[E][a] = [G][a] \qquad \text{... 2-5}$$

$$\text{where} \quad [G] = pinv([F])[E] \qquad \text{... 2-6}$$

Critically, matrix $[G]$ only has to be computed once so it is prudent to select many frequency points to optimize the accuracy spread over the required bandwidth of the spectral matching process. Because longer output bitframes achieve better spectral matching, there is a trade-off against the computational overhead, accuracy and time dispersion. In practice output bitframes of 32 or 64 samples yield excellent results. Also, because output bitframes are longer than input bitframes, a vertical overlap-and-add process is used to combine adjacent bitframes as depicted in Figure 3; the input bitframes are concurrent while the output bitframes must overlap. This process is defined as pseudo non-linear convolution because output bitframes are individually calculated and are not subjected to a fixed, time-invariant filter response extending over many hundreds of samples; here bitframes are typically just 8, 16 or 32 bit in length so because of complex samples, time dispersion is intrinsically low.

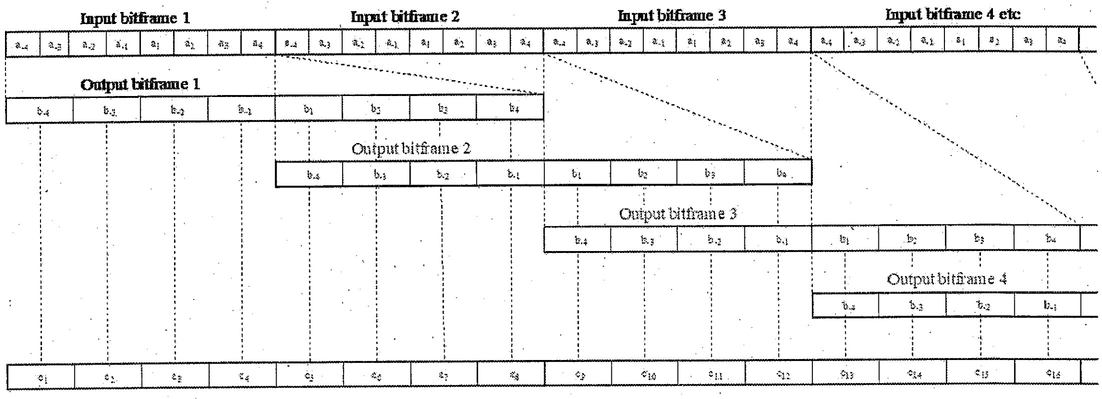


Figure 3 Vertical convolutional-addition of overlapping *b*-bitframes to form output data stream.

3 BAND SPLITTING USING DISCRETE HILBERT TRANSFORM (DHT) AND COMPLEX SAMPLES

Following DSRC to an 88.2 kHz sampling rate but prior to encoding, the Discrete Hilbert Transform (DHT) [6] and a matrix procedure are used to decompose the complex data DC into two band-limited LPCM streams representing ABB data 0 to 22.05 kHz and UAB data 22.05 kHz to 44.1 kHz respectively, thus allowing both streams to be decimated to a sampling rate of 44.1 kHz. Since all algorithms in the prototype software were written in *Matlab*, then for conciseness a pseudo-*Matlab* notation is employed where the process is as follows:

The complex data vector DC has the form, $DC \equiv real(DC) + j * imag(DC)$ ... 3-1

Generating the weighed Hilbert impulse response *h* using *Matlab* operator *hilbert*,

$$h = j * imag\left(hilbert\left(\overline{U}\right).* win\right) \qquad \text{... 3-2}$$

where $\overline{U}$ is a unit impulse vector and *win* is a weighting vector to time-constrain *h*. Vector DC is then converted into a real data vector DO by convolution with *h*, where because of finite-length vector processing, is performed in the frequency domain as,

$$DO = real(DC) - ifft\left(fft\left(imag(DC)\right).* fft(h)\right) \qquad \text{... 3-3}$$

To band split DO centred on 22.05 kHz according to ABB and UAB, the DHT is used to derive a vector DT whereon the UAB band is inverted compared to DO, i.e.

$$DT = ifft\left(fft(DO).* fft(h)\right) \qquad \text{... 3-4}$$

A complex variant of vector DO can then be formed using a sum-and-difference procedure whereby the real samples correspond to the ABB signal and the imaginary samples to the UAB signal, i.e.

$$DO \Rightarrow (DO + DT)/2 + j(DO - DT)/2 \qquad \text{... 3-5}$$

Decimation from 88.2 kHz to 44.1 kHz is achieved simply by discarding alternate complex samples. However, because UAB was located above the new sampling frequency of 44.1 kHz, its spectrum is now reversed because of folding; this can be corrected in the time domain by inverting alternate imaginary-sample components in DO. Thus defining vector $rev = [1,-1,1,-1,...]$ having the same length as DO,

$$DO \Rightarrow real(DO) + j * rev. * imag(DO) \qquad \text{... 3-6}$$

Hence ABB and UAB signals are now combined into a complex form suitable for encoding.

4 ENCODER ARCHITECTURE AND SIGNAL PROCESSING

This Section describes an overview of the encoder architecture depicted in Figure 4 where the initial stages of DSRC and band-splitting were described in Sections 2 and 3. It is assumed for purposes of this discussion that the final encoder output *comp* is sampled at 44.1 kHz with each 24-bit sample sub-divided into fields of 16 Most Significant Bits (MSBs) and 8 Least Significant Bits (LSBs) respectively. Alternative sampling rates and sample decomposition are of course possible including the use of 32-bit data and the size of data chunk assigned to the UAB channel.

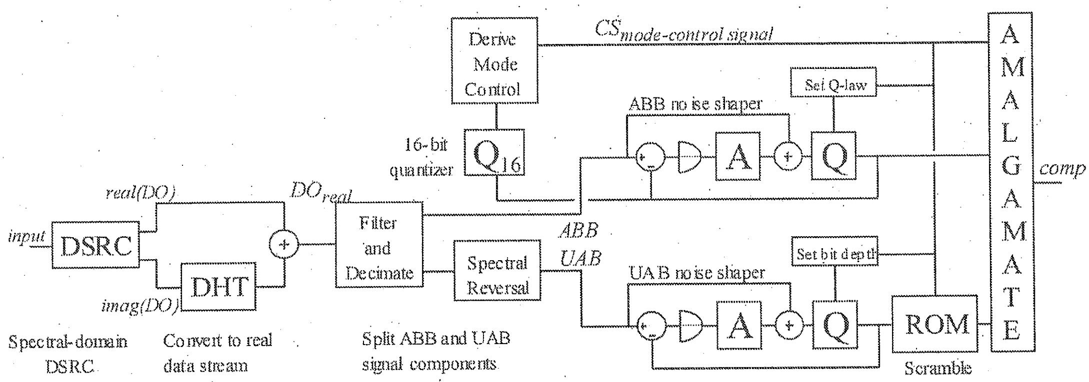


Figure 4 Encoder architecture and sub-processes.

Paramount to the encoder (and subsequent decoder) is that 16 bit of the ABB data chunk transforms directly to the 16 MSBs of the output data without modification, while some, or all, of the 8-LSB chunk can be assigned dynamically to augment the audio-band data to enable up to full 24-bit precision when the ABB signal is of a sufficiently low level. The dynamic bit assignment of LSB chunk takes its cue from a control vector CS that is derived only from the 16 MSB ABB chunks which by design must be available at both the encoder and decoder. If processing sample r then CS is derived from sample r-1 and is formed from the logarithm of the quantized average level of the two stereo signals. Using the past sample allows the option for noise shaping and dither to be applied as the same data is then available for the decoder. Consequently, the dynamic state of the system is always known allowing both encoder and decoder to track and therefore to know precisely the assignment of each data chunk within each 24-bit word. This means that as the signal level falls within core 16-MSB chunks, then LSB chunks can progressively reduce in size from N = 8 to 0 bit orchestrated by the mode-control vector and a set of pre-defined decision thresholds. Hence, ABB data is assigned 16+N bits and the UAB data N bits. The assignment of N is derived from CS which controls a set of comparators where example transition values for a 24-bit composite signal are:

N=8;		% mode 6 operation (maximum resolution for UAB coded signal)
N=7;	mode65_24 = 30	% mode 6 to mode 5 transition, dB threshold below 0 dB
N=6;	mode54_24 = 50	% mode 5 to mode 4 transition, dB threshold below 0 dB
N=5;	mode43_24 = 60	% mode 4 to mode 3 transition, dB threshold below 0 dB
N=4;	mode32_24 = 70	% mode 3 to mode 2 transition, dB threshold below 0 dB
N=3;	mode21_24 = 80	% mode 2 to mode 1 transition, dB threshold below 0 dB (no effect < -90.309 dB)
N=0;		% mode 1 operation (no UAB coded signal, noise substitution)

Note for different composite word bit depths including mono UAB operation, different thresholds can be assigned. However, encoder and decoder must operate with the same comparator thresholds.

An example of CS (*see black trace*) together and mode-control thresholds (*see red lines*) defined by the above table are shown in Figure 5, derived using a short music sequence.

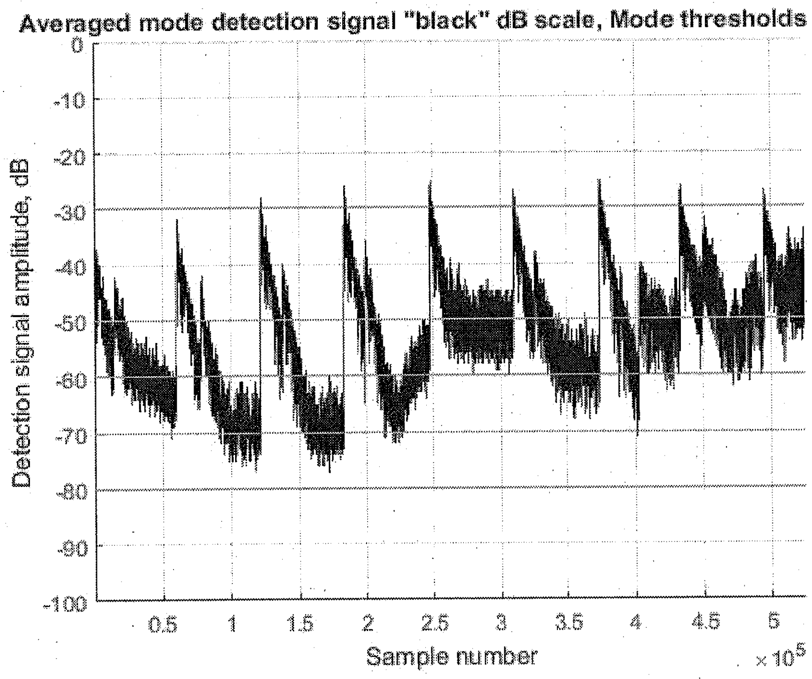


Figure 5 Mode-control signal *CS* derived from the 16-MSB ABB data chunks.

The UAB signal is then extracted from the imaginary samples in DO and applied to a logarithmically companded, 2nd-order noise-shaping algorithm. The logarithmic quantizer in the noise shaper is dynamically scaled via the mode-control signal CS to match the data precision assigned to the current LSB chunk where inevitably the quantization distortion is greater than that within the ABB signal. Dynamic data assignment implies that as the ABB signal falls, the quantization accuracy of the UAB channel degrades still further, although in mitigation its subjective significance becomes vanishingly small for low-level ABB signals. In the limit this enables the UAB data to be muted and seamlessly substituted with benign, spectrally shaped low-level noise. Also, because the UAB signal has been frequency shifted downwards to the 0 to 22.05 kHz band prior to noise shaping, then following the decoder, quantization distortion is reduced effectively in the 22.05 kHz to ≈25 kHz band, yielding lower artefacts in the frequency range just above the principal audio band.

An additional process is applied to the UAB data using N-related pseudo-random sequences so the UAB data chunks act as benign noise in the composite output signal. Such low-level noise within the composite signal can benefit a DAC by aiding the de-correlation of errors arising from minor differential DAC non-linearity. A second-order noise shaping option was also included to enhance re-quantization in the ABB channel especially as the 16+N bit quantizer resolution adapts on a sample-by-sample basis as a function of CS.

Encoding is then completed by amalgamating the 16+N bit ABB and N bit UAB data chunks into a composite 24 bit, 44.1 kHz LPCM stream, a process requiring the assembly of data frames using an overlap-and-add procedure to eliminate frame-based artefacts such as those arising from circular convolution with the DHT.

5 DECODING WITH UAB RECONSTRUCTION

The decoder architecture required to reconstruct the wide-band output signal is shown in Figure 6 where it is assumed no processing has been applied to the encoder output signal, *comp*.

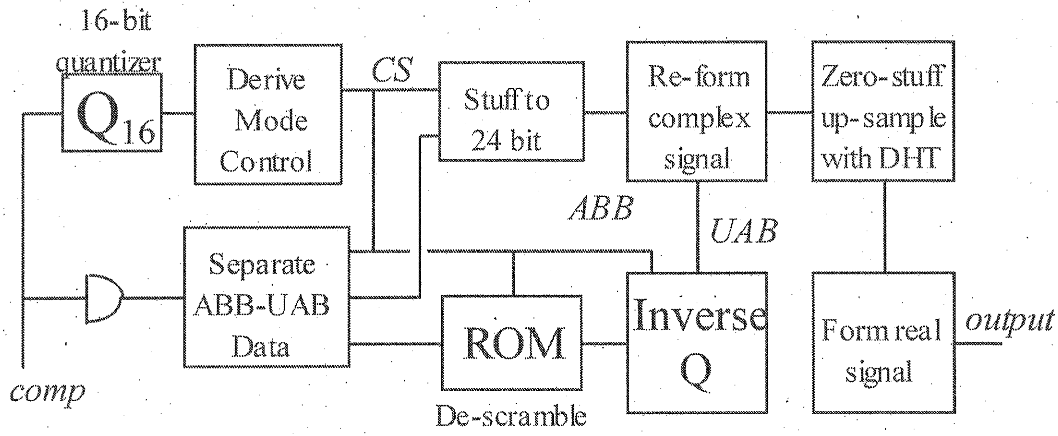


Figure 6 Decoder architecture and sub-processes.

To recover both the embedded 16+N bit ABB and N-bit UAB data chunks from each sample of *comp*, the mode-control vector CS must be reformed from the 16-MSB ABB data again using comparators to identify the sample-specific modes which define data chunk size N. Also, since the UAB data was scrambled in the encoder, mode-specific inverse look-up tables are assigned according to N to re-form the true sample values and because logarithmic quantization was employed in the UAB noise shaper, a complementary weighting characteristic is applied to scale the output data. A complex vector DOD is then formed where $real(DOD)$ is the ABB signal and $imag(DOD)$ is the UAB signal, i.e.

$$DOD \Rightarrow real(DOD) + j * rev.* imag(DOD) \qquad \ldots 5.1$$

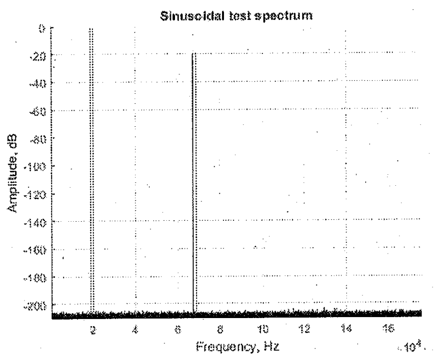
Note that in constructing DOD, alternate samples of the UAB signal are inverted by vector *rev* to facilitate spectral reversal, mirroring the reversal performed within the encoder; each ABB data chunk is then padded to 24 bit using randomly generated bits to enhance low-level DAC performance. Finally, the sampling rate is increased to 88.2 kHz by zero-stuffing alternate samples of DOD and applying the DHT together with a sum-and-difference matrix to filter spectral replicates and reform a real broad-band decoder output signal, i.e.

$$DOD \Rightarrow real(DOD) + imag(DOD) + ifft\big(fft\big(real(DOD) - imag(DOD)\big).* fft(h)\big) \qquad \ldots 5\text{-}2$$

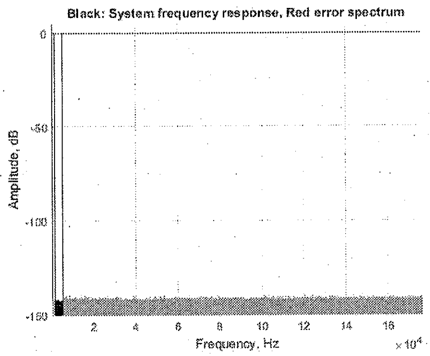
Since all signal processing is performed frame-by-frame, then frame overlap-and-add is employed to construct a continuous audio output *.wav* file in a 24-bit, 88.2 kHz format.

6 PERFORMANCE EXAMPLES

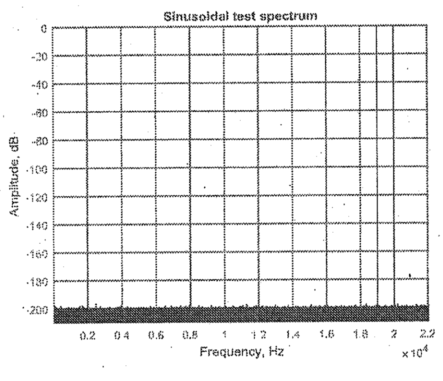
A performance profile is presented to illustrate how accurately the overall system is able to maintain signal integrity, especially in frequency regions where human hearing is most sensitive. The first set of test results are shown in Figure 7 and demonstrate only the DSRC using both tonal and filtered noise test sequences. The down-converted 4-tone test reveals an extremely low noise floor with no inter-modulation distortion evident while the flat amplitude-equalized noise signal, which includes a band-stop region, confirms an excellent frequency response (error spectrum in red) with no significant rise in noise within the stop-band region arising from non-linearity.



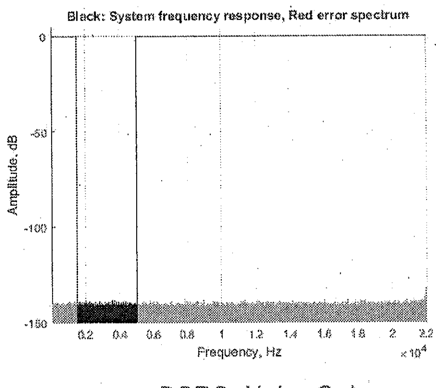
4-tone: 19 kHz, 20 kHz, 67.3 kHz, 69 kHz



Noise, 1.5 kHz to 5 kHz band-stop filter

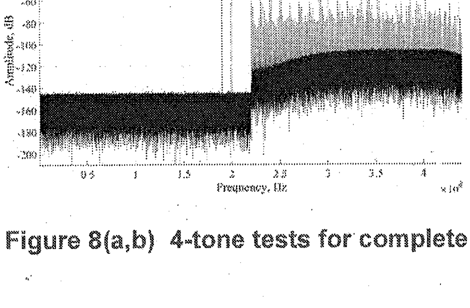


DSRC: 19 kHz and 20 kHz



DSRC: Noise Only

Figure 7 DSRC performed on 4-tone and band-stop filtered noise signals.



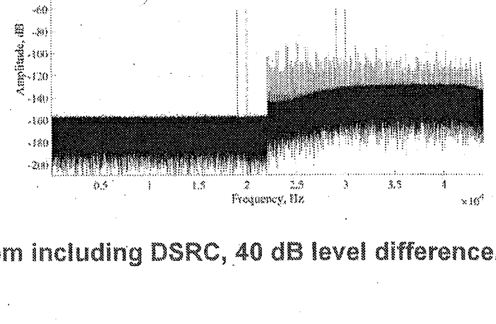


Figure 8(a,b) 4-tone tests for complete system including DSRC, 40 dB level difference.

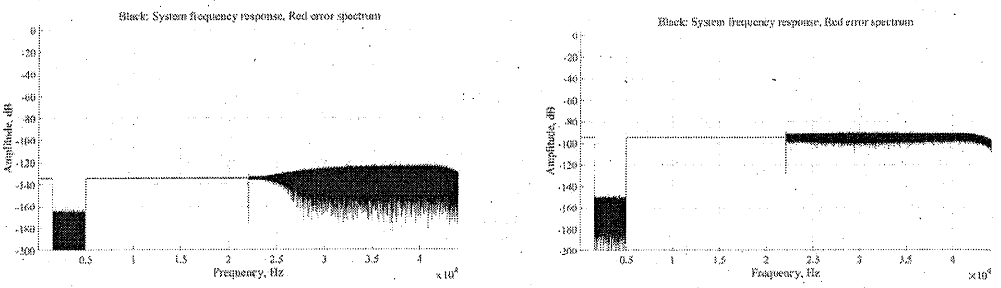
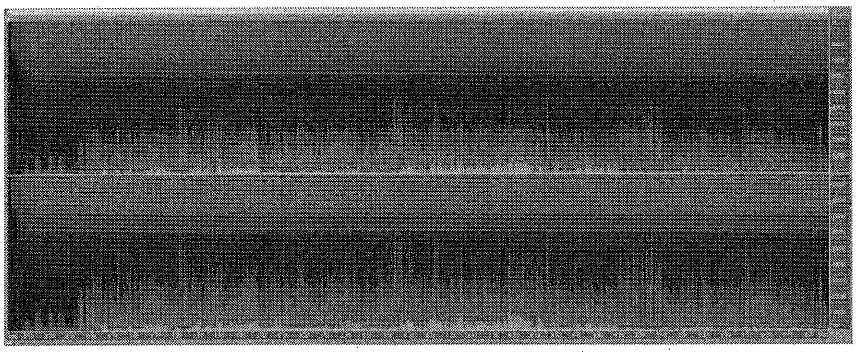
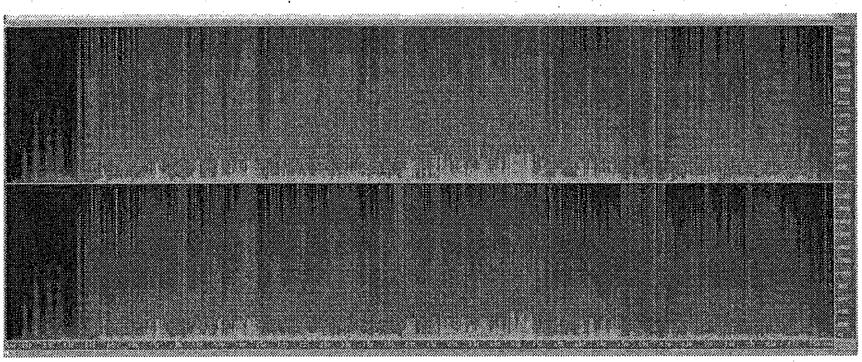


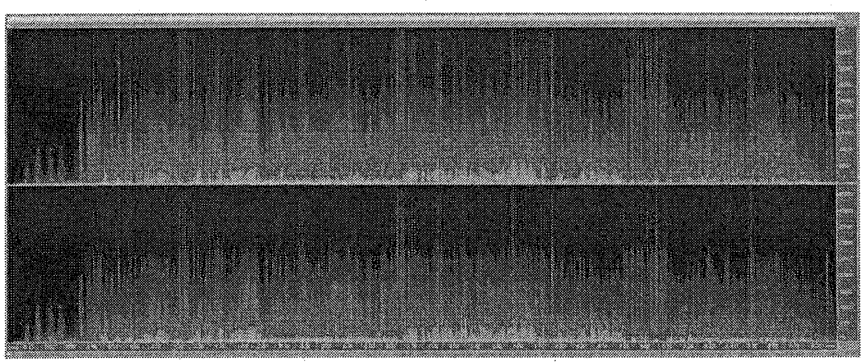
Figure 9(a,b) Noise tests for complete system including DSRC, 40 dB level difference.



(a) Spectra of music sequence: source file, 24 bit @ 192 kHz.



(b) Spectra of music sequence: encoded file, 24 bit @ 48 kHz.



(c) Spectra of music sequence: decoded file, 24 bit @ 96 kHz.

Figure 10(a-c) Spectra of music sequence: (a) source, (b) coded and (c) decoded.

Next, two sets of spectral analysis are performed on the complete encoder-decoder with results for a multi-tone test shown in Figure 8(a,b) and for noise test shown in Figure 9(a,b), where result pairs differ in level by 40 dB. The tests reveal low noise and distortion for the critical audio band as anticipated while the ultrasonic band displays additional noise and distortion products due to its much lower coder resolution. However, because of dynamic bit assignment, a lower noise floor can be observed when ABB signal level is lower. Finally, Figure 10(a-c) displays stereo spectra for a music signal (*source 192 kHz, coded 48 kHz and decoded 96 kHz*) where the ultrasonic band although noisier, contains a fair replica of the signal components present in the source signal.

7 CONCLUSIONS

The paper has presented a coding method enabling the ultrasonic signal components within an audio signal to be separated and buried within a LPCM stream, typically 24 bit, 44.1 kHz. The principal aim is to achieve minimal signal processing within the audio band to facilitate high transparency although accepting degradation within the ultrasonic band is a valid compromise due to its lower subjective significance. The design retains a high degree of compatibility between the coded signal and a standard LPCM signal to allow playback without decoding. This is achieved by retaining the 16 MSBs without processing while ensuring the 8 LSBs carrying buried data are randomized by complementary, pseudo-random look-up tables to help reduce distortion from DAC differential non-linearity. Dynamic assignment of the 8 LSBs is also employed to enhance low-level ABB resolution, albeit at the expense of UAB resolution, since at lower signal levels the ultrasonic advantage becomes vanishingly small. This advantage is retained in decoding but with the benefit of bandwidth extension commensurate with a doubling in sample rate. Noise shaping is employed in both ABB and UAB channels to improve linearization in the presence of the quantization law modulation resulting from dynamic bit allocation. Coding is supplemented by the spectral-domain DSRC algorithm as this offers extremely low in-band conversion artefacts as demonstrated by simulation in Section 6. This paper has discussed a 24-bit data format although this can be extended to 32 bit allowing full 24-bit resolution for the ABB signal. Finally UAB coding artefacts can be reduced by mixing the UAB signal to mono and using stereo-like envelope steering.

If you wish to audition the results of this class of spectral-domain DSRC using your own LPCM (.wav) or DSD (.dsf) source tracks, you may contact the author by email: mjh@essex.ac.uk. In return, comments on performance would be welcomed in order to gain broad opinion on this class of DSRC pitched against conventional methods of conversion.

8 REFERENCES

1. Lagadec, R., "Digital Sampling Frequency Conversion", *Digital Audio*, Audio Engineering Society, pp. 90-96, New York, 1982
2. Application of Spectral-Domain Matching and Pseudo Non-Linear Convolution to Down-Sample Rate Conversion (DSRC), Hawksford, MOJ, 101st Convention of the Audio Engineering Society, Los Angeles, USA, November, 1996, Paper 4362 (J-2)
3. Extended-Definition Digital Audio Systems for High-Capacity CD, Hawksford, MOJ, IEE Colloquium on Audio Engineering, pp 2-1 to 2-12, 1st May 1995, Digest No. 1995/089
4. *Invited Paper:* Multi-channel High Definition Digital Audio Systems for High-Density Compact Disc, 143rd Convention of the Audio Engineering Society, New York, USA, October, 2017, Paper 9811
5. Craven, P. G., "Antialias Filters and System Transient Response at High Sample Rates," *J. Audio Eng. Soc.*, pp. 216–242
6. Gold, B., Oppenheim, V. and Rader, C.M., "Theory and Implementation of the Discrete Hilbert Transform", (Lincoln Laboratory, Massachusetts Institute of Technology), *Symposium on Computer Processing in Communications*, Polytechnic Institute of Brooklyn, April 8-10, 1969
7. Craven, P. G., "Antialias Filters and System Transient Response at High Sample Rates," *J. Audio Eng. Soc.*, pp. 216–242
8. Hawksford, M.O.J., "System Measurement and Identification using Pseudorandom Filtered Noise and Music Sequences", *J. Audio Eng. Soc.*, vol. 52, no. 4, pp. 275-296, April 2005