# Some Notes on Digital Audio Topics

**MARTIN WILLCOCKS**

*Willcocks Research Consultants, Santa Monica, CA*

**0. INTRODUCTION:** These notes were originally presented informally to the second AES meeting on digital audio in Atlanta, but have been revised extensively before publication here. The topics include sampling frequency considerations for video-based recording systems, linear and nonlinear source encoding, interfacing between equipment, and sampling rate conversion.

## 1. SAMPLING FREQUENCY: COMPATIBILITY WITH VIDEO-BASED SYSTEMS

Several current digital recording systems use video tape recorders and encode the digital information into a pseudo-video signal to ensure proper operation of the video tape recorder (VTR). Each sample is represented by an $n$-bit digital word, an integral number $a$ of such words being grouped in each active line period, and an integral number $b$ of active lines being recorded in each field. For error-correction purposes, each word may be recorded twice at different locations, or a parity word formed by the exclusive-OR of corresponding left-channel and right-channel samples may be recorded, the three words in this case being spatially distributed by interleaving. Assuming two-channel recording with redundancy factor $K = 2$ for double recording or $K = 1.5$ for parity word recording, the sampling rate is related to the horizontal frequency $f_H$ and the number of lines per frame $N$ by

$$f_s = \frac{ab}{K} (f_H/N).$$

The horizontal frequency $f_H$ is 15.625 kHz, 15.75 kHz, or 15.734264 kHz in the 625-line PAL/SECAM, 525-line monochrome, or 525-line NTSC color TV standards, respectively. The active line period is a minimum of 52.1 $\mu$s or 0.82 H in the 525-line monochrome standard, and is marginally longer in the other standards. The 4.2-MHz video bandwidth of the 525-line system sets an upper limit of about 440 bits per active line, but bandwidth limitations of video cassette recorders suggest a more practical limit of about 220 bits per line, corresponding to fourteen 16-bit words or seventeen 13-bit words. Even this recording density is uncomfortably high for some consumer video cassette recorders (VCR) [1].

The vertical sync waveform occupies nine lines per field, but as head switching occurs from five to ten lines before the field sync pulses [2], 11 to 16 lines per field are unusable for data, leaving between 246½ and 251½ active lines per field for the 525-line systems or from 296½ to 301½ active lines per field for the 625-line systems. The number of active lines used should be a multiple of five in the 525-line systems and the same multiple of six in the 625-line systems for corresponding sampling rates.

Because of bandwidth limitations, most of the PCM video-based systems so far announced use parity word recording ($K = 1.5$) with nine 13-bit words plus a cyclic redundancy check code word (CRCC) on each active line, with 245 active lines per field in the 525-line format or 294 active lines per field in the 625-line format. This results in

a sampling frequency of 44.05594 kHz for the 525-line NTSC system, or 44.1 kHz for the 525-line monochrome and 625-line standards.

For convenience, the system clock frequency should be a multiple of both the sampling rate and the horizontal frequency in any video-based recording system, but need not be identical in the 525-line and 625-line systems. Suitable clock frequencies can be deduced, given the number of bits required per active line period and both sampling and horizontal frequencies. A suitable clock frequency for these systems for the 525-line NTSC standard is 2.4230769 MHz, or $55 f_S$, which is also $154 f_H$. This would allow at least 126 bits per line, and requires a video bandwidth of at least 1.2 MHz, but preferably more.

Table I gives some possible video-based formats having sampling rates close to those suggested by Heaslett [3]. In addition to formats with integer numbers of words per line and active lines per field, it is possible to have formats with half-integer numbers. Values of $a$ are given for both double recording ($K = 2.0$) and parity word recording ($K = 1.5$). The 0.1% difference between the horizontal frequencies of the monochrome and color 525-line standards results in a corresponding difference between sampling frequencies, but techniques exist for interpolating enough extra samples over a number of fields to compensate for this difference.

Although Table I is not comprehensive, formats are shown for all the principal sampling rates suggested by Heaslett, as well as the 50.4-kHz rate suggested by Doi [4] and independently by Gibson [5] at the Los Angeles meeting of the AES on digital audio in April.

The 50.0-kHz sampling rate is hard to fit into a simple video-based format; because in parity word recording three words are recorded for each stereo sample pair, at ten words per line a subblock of three lines is needed before the format recurs. As each field contains 250 active lines, a block of three frames is required before the data format is repeated. Recording at the higher density of 12 words per line overcomes the first problem; however, the block length required is still three frames, with 208, 208, and 209 lines used for data in each three successive fields, or even 208 1/3 lines per field.

A frequency such as 52.5 kHz or 50.4 kHz requires only two-line subblocks, and as there is an even number of lines in the field, the data format in each field is the same. Because of head-switching limitations in some VCRs, 52.5 kHz may not be suitable, as it requires 250 lines of data per field. 50.4 kHz is in a simple 8:7 ratio to 44.1 kHz (but the ratio to 44.05594 kHz is 143:125), which may make digital sampling frequency conversion easier.

## 2. ABSOLUTE PITCH ACCURACY

Heaslett contends that the 0.1% pitch error resulting from the same difference between NTSC and monochrome 525-line standard horizontal frequencies is unacceptable, and he introduced the principle of variable latency between block and frame boundaries to overcome this problem [3]. The real problem is the need to synchronize sound and picture over long periods of time, rather than the absolute pitch error, which could probably be detected by a musician but not by the average listener.

When movies are televised, the 24-Hz frame rate may be converted to the 29.97-Hz rate needed for telecine by interpolating or repeating one frame in five, and by slowing the film 0.1% to take care of the difference between 30 Hz and 29.97 Hz. This results in a pitch error of 0.1%, but does not upset the synchronization between picture and sound. Unlike the 4.167% pitch error which resulted in the early days of telecine in England, when movies were speeded up from 24 to 25 frames per second, and which decidedly changed the vocal timbre of such characters as Perry Mason and the Lone Ranger, this small error is not noticeable. As pitch error of this magnitude or larger can occur from the use of synchronous motors or

Table I. Some possible video-based formats for digital audio recording.

| Number of Words per Line $a$ | | Number of Active Lines per Field $b$ | Sampling Rate $f_S$ 525-Line System NTSC Color | (kHz) Monochrome | Number of Active Lines per Field $b$ | Sampling Rate $f_S$ 625-Line System |
|---|---|---|---|---|---|---|
| $K = 2$ | $K = 1.5$ | | | | | |
| 12 | 9 | 245 | 44.05594 | 44.100 | 294 | 44.100 |
| 12 | 9 | 250* | 44.95504 | 45.000 | 300* | 45.000 |
| 13 | | 245 | 47.72727 | 47.775 | 294 | 47.775 |
| 13 | | 246 | 47.92208 | 47.970 | 295 | 47.9375 |
| 13 | | 247* | 48.11688 | 48.165 | 296 | 48.100 |
| | 10 | 240 | 47.95205 | 48.000 | 288 | 48.000 |
| | 10 | 250* | 49.95005 | 50.000 | 300* | 50.000 |
| 14 | 10½ | 237 | 49.72028 | 49.770 | 285 | 49.875 |
| 14 | 10½ | 238 | 49.93007 | 49.980 | 286 | 50.050 |
| 14 | 10½ | 240 | 50.34965 | 50.400 | 288 | 50.400 |
| 14 | 10½ | 250* | 52.44755 | 52.500 | 300* | 52.500 |
| 15 | | 225 | 50.57442 | 50.625 | 270 | 50.625 |
| 15 | | 240 | 53.94605 | 54.000 | 288 | 54.000 |
| 15 | | 250* | 56.19381 | 56.250 | 300* | 56.250 |
| 16 | 12 | 200 | 47.95205 | 48.000 | 240 | 48.000 |
| 16 | 12 | 208 | 49.87012 | 49.920 | 250 | 50.000 |
| 16 | 12 | 209 | 50.10988 | 50.160 | | |
| 16 | 12 | 225 | 53.94605 | 54.000 | 270 | 54.000 |
| 16 | 12 | 250* | 59.94006 | 60.000 | 300* | 60.000 |

* May be incompatible with some VCRs.

stroboscopes using the mains frequency as a reference, when recording or playing disc records, it is questionable whether absolute pitch accuracy is essential in video-based digital recording systems for music.

## 3. LINEAR AND NONLINEAR CODING AND TRANSCODING

In most computer systems, twos complement coding is used to represent numbers. Zero is represented by the 16-bit word 0000 0000 0000 0000 and negative numbers by the twos complement of the corresponding positive numbers. Thus +1 is represented by 0000 0000 0000 0001 and −1 by 1111 1111 1111 1111. Analog-to-digital converters normally use binary offset coding, which can be converted to or from twos complement code by complementing the first bit. In this code, −1, 0, and 1 are represented by 0111 1111 1111 1111, 1000 0000 0000 0000, and 1000 0000 0000 0001, respectively. Actually, each word represents a small range of analog values because the signal is quantized, so it would be better to think of the digital word 1000 0000 0000 0000 representing the range from just above 0 to just below +1, or the central value +½ with uncertainty of ±½ l.s.b. units. A 16-bit linear code has a range of ± 32,768 l.s.b. units, and an estimate of the quantizing noise, assuming the error to be uniformly distributed, is $1/2\sqrt{3}$ l.s.b. units, which leads to an estimate of the dynamic range and the instantaneous signal-to-quantizing noise ratio of 98.1 dB.

Nonlinear codes in practice approximate to various theoretically continuous companding laws by defining a number of segments of the total range, and subdividing each segment linearly into a number of levels, the difference between two adjacent quantizing levels being either a binary fraction of the maximum level in the segment or a binary fraction of the range covered by the segment. The number of segments is usually a power of 2 (although the scale of the two middle segments is the same as they are symmetrically disposed around the analog zero point), and the segment number, or exponent, is an $x$-bit binary word if there are $2^x$ segments. The mantissa, which identifies the level within a segment, is an $m$-bit binary word. The most economical kind of code is illustrated in Fig. 1, in which the ratio between the maximum levels in any two adjacent segments except the middle pair is a number $r$. The scale factor of the segments also changes in this ratio, except that the scale of the two segments adjacent to the middle pair is $r - 1$ times the middle segment scale. The range of values that the code can represent is $\pm R$, where

$$R = 2^m r^{(2^{x-1}-1)}.$$

This is a polygonal companding code. When $r = 2$, it can be described as a binary-scaled polygonal companding code. A code of this type, having an 8-bit mantissa and a 4-bit exponent, is used in the TEAC 4-channel PCM processor [6].

Considering a binary-scaled polygonal code with $m = 8$ and $x = 4$, the range given by this equation is ± 32 768 l.s.b. units of the lowest range, identical to the 16-bit linear code. The dynamic range is therefore also 98.1 dB, but the instantaneous quantizing noise is now larger, and depends on the signal level. An estimate of the quantizing noise or distortion at maximum level for this type of code is $0.69 \times 2^{-m-2} \times 100\%$, in this case 0.0675%. As the signal is reduced, the distortion and noise percentage increases, reaching a maximum at somewhat over half the maximum level, then falling to about the same value at 6 dB below maximum. The distortion and noise expressed as a percentage of signal shows a characteristic scalloped curve as given in the TEAC literature [6].

It is very easy to transcode between this code and the 16-bit linear code, as explained below. Similar codes, such as a 12-bit mantissa and 3-bit exponent binary-scaled polygonal companding code, can also be easily transcoded to 16-bit linear.

Another kind of nonlinear code which is easier to implement in a practical system uses a switched attenuator before the analog-to-digital converter, and changes the range by switching in more attenuation whenever the maximum range of the converter would be exceeded. Most of the Japanese consumer PCM systems are reported to be of this type, with a 12-bit mantissa, a single-bit exponent, and a ranging factor of either 4 [7] or 8 [8]. Provided that the range factor is a power of 2, this type of code can also be converted to or from a 16-bit linear code without too much difficulty. As an example of transcoding between linear and nonlinear codes, see Table II.

In the central segment, the nonlinear 8+4 code and the 16-bit linear code are both linear and have the same scales and the same central values. The last eight bits of the 16-bit word for positive values of the signal form the mantissa of the 8+4 code representation. In the next segment, the first level has central value 513, corresponding to the two 16-bit levels 512½ and 513½, or the range of analog values from 512 to 514 l.s.b. units, and the mantissa is given by the eight bits immediately following the leading 1 of the magnitude part of the 16-bit code. In the next segment, the first level is 1026, corresponding to

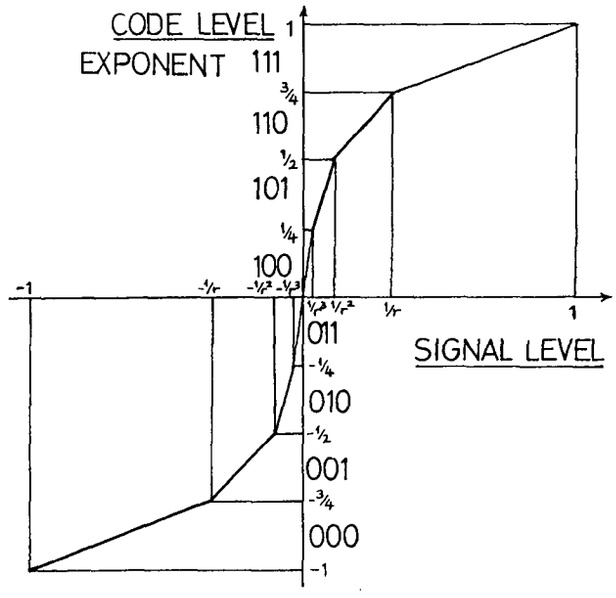


Fig. 1. A constant-ratio nonbinary ranging nonlinear code.

the range of analog values from 1024 to 1028 units, and the mantissa is again represented by the eight bits following the leading 1 of the 16-bit code. For negative signal polarity, the 16-bit code is first complemented to find the magnitude, from which the mantissa of the 8+4 code can be determined in the same way.

Reverse transcoding can be achieved similarly. Since the bits after the eight-bit mantissa portion of the 16-bit words generated may take any value, the most satisfactory way of obtaining the correct central value while masking the quantizing errors is to make all the bits after the mantissa either 0 or 1 with equal probability, using a random generator. The signal-to-noise ratio will not be affected by this, since it is assumed that the quantizing noise has the same magnitude as the noise artificially inserted into the 16-bit code thus generated.

## 4. CODE FORMAT WITHIN A 16-BIT SLOT

In the case of the 16-bit linear code, this is just the 16-bit word. For a 12+3 format, a parity bit could be included to give, say, even parity. For a 12+1 format, three following bits could give overall odd parity, one of the three could designate whether a 12-dB (× 4) or 18-dB (× 8) range factor was used, and another could designate whether preemphasis was used (assuming a standard preemphasis curve). For the 8+4 format, as the 13th bit is a parity bit, the remaining three bits could be assigned to designate preemphasis, and establish overall odd parity, one bit being unassigned. With this kind of code format, it would be possible to identify the recording code unambiguously from the recovered data, so that automatic switching of the decoding mode could be accomplished.

## 5. DIGITAL INTERFACE ON A WIRE

When interconnecting equipment, there should be a provision for correcting any possible transmission errors, but the error-correcting scheme does not need to be as redundant as those used for tape recording or video disk records, as the probability of long error bursts is much lower. The minimum length of an error-correcting code that will detect double errors and correct single errors in a 16-bit word is 5 bits; therefore a total word length of 21 bits would be sufficient to protect individual words against single errors. Another approach might be to add a cyclic redundancy check code to the 16-bit data words, in which case at least four bits would be required to be reasonably sure of detecting errors, but this would not necessarily correct all errors. Whatever scheme is adopted, the "slot" for digital interface on a wire should be at least 21 bits long, plus some synchronizing code.

In Table III some possible formats for self-identifying codes are suggested, both for interface on a wire and for format in a 16-bit slot. Mantissa bits are denoted by M, exponent bits by X, even parity bits by E, odd parity bits

Table II. Transcoding between linear and nonlinear codes.

| 16-Bit Linear Code | | | 8+ 4 Nonlinear Code | |
|---|---|---|---|---|
| Digital Representation | Central Value | Limit Value | Central Value | Digital Representation |
| 1000 0001 1111 1111 | 511½ | − 511 | −511½ | 1001 1111 1111 |
| 1000 0010 0000 0000 | 512½ | − 512 | − | |
| 1000 0010 0000 0001 | 513½ | − 513 | 513 | 1010 0000 0000 |
| 1000 0010 0000 0010 | 514½ | − 514 | − | |
| 1000 0010 0000 0011 | 515½ | − 515 | 515 | 1010 0000 0001 |
| mantissa | | − 516 | − | |
| 1000 0011 1111 1111 | 1023½ | −1023 | 1023 | 1010 1111 1111 |
| 1000 0100 0000 0000 | 1024½ | −1024 | − | |
| 1000 0100 0000 0001 | 1025½ | −1025 | | |
| 1000 0100 0000 0010 | 1026½ | −1026 | 1026 | 1011 0000 0000 |
| 1000 0100 0000 0011 | 1027½ | −1027 | | |
| mantissa | | −1028 | − | |

Table III. Self-identifying digital code formats.

| Code | Word Format (16 bits) | Bit Format on Wire (21 bits) |
|---|---|---|
| 16-bit linear (flat) | MMMM MMMM MMMM MMMM | MMMM MMMM MMMM MMMM CCCCC |
| 12+3 binary-scaled polygonal companding | XXXM MMMM MMMM MMME | XXXM MMMM MMMM MMME CCCCC |
| 12+1 12 dB ranging (flat) | XMMM MMMM MMMM MØØØ | XMMM MMMM MMMM MØØØ CCCCC |
| 12+1 12 dB ranging (with standard preemphasis) | XMMM MMMM MMMM MØ1Ø | XMMM MMMM MMMM MØ1Ø CCCCC |
| 12+1 18 dB ranging (flat) | XMMM MMMM MMMM M1ØØ | XMMM MMMM MMMM M1ØØ CCCCC |
| (preemphasized) | XMMM MMMM MMMM M11Ø | XMMM MMMM MMMM M11Ø CCCCC |
| 8+4 binary scaled (flat) polygonal companding | XXXX MMMM MMMM EØØ1 | XXXX MMMM MMMM EØØ1 CCCCC |
| (preemphasized) | XXXX MMMM MMMM EØ1Ø | XXXX MMMM MMMM EØ1Ø CCCCC |

by 0, correction code bits by C, and zero and unity values by 0 and 1, respectively.

In Table III codes are suggested for identifying both flat and preemphasized coding schemes, it being assumed that where preemphasis is used, it will conform to a standard curve, and preferably one that can be easily deemphasized in the digital domain. The use of even parity to detect a 12+3 code, which has only one available bit for such purposes when fitted to a 16-bit slot, necessitates the use of odd parity to detect all other codes (unless more elaborate detection schemes are used; however, simplicity is the key to efficient self-detecting codes). The unassigned bits are then contrived to give direct information on the actual code in use, so that equipment can if required accommodate more than one code, and switch automatically to the appropriate decoding mode.

It is suggested that data on a wire be serial, self-clocking, and error-correcting using a standardized scheme. Cable and terminations should be in accordance with current practice in video interconnections, for example, using 50-Ω impedance of 75-Ω impedance and BNC or multiple coaxial connectors, and signal levels should be compatible with TTL or CMOS.

## 6. SAMPLING RATE CONVERSION

Since it appears likely that more than one standard sampling rate may be recommended, the problem arising is digital transcoding between linear and nonlinear codes, discussed above, and conversion between different sampling rates. Provided both sampling rates are above the Nyquist rate, it is possible to do this conversion without any loss of information.

Where this has been required to date, the digital signal has been converted back to analog form and then resampled at the new rate; because of limitations in the accuracy of conversions it would be preferable to do this entirely in the digital domain.

Ideal digital-to-analog conversion requires that the sample values be generated as ideal impulses, of infinitesimal duration but with time integral equal to the sample values, and applied to an ideal low-pass filter, the signal at the filter output being a perfect reconstruction of the original analog signal before sampling. However, ideal impulses are not easy to approximate, and ideal low-pass filters are physically unrealizable because they have an infinite delay between input and output, so a practical digital-to-analog conversion involves some distortion (not necessarily nonlinear distortion). When resampled at a new rate, there will be some loss of information as a result of this distortion, which could show as unwanted sidebands, departures from flat frequency response, phase errors, etc.

Changing the sampling frequency in the digital domain is entirely equivalent to interpolating digitally between samples. We were surprised to find references on this subject as early as 1915, and the papers referred to below contain many useful references to this and allied subjects [9–11].

Basically, there is an error inherent in the process of determining an intermediate ordinate from a finite number of samples, which can be minimized by choosing a suitable weighting function, or window, so that the samples nearest to the interpolation point have the most effect on the interpolated value, but the largest feasible number of samples is used in the interpolation process. There is therefore a delay in generating the interpolate, which one would expect to be commensurate with that involved in the practical low-pass filter used in digital-to-analog conversion, for the same error. The choice of a suitable window would be critical in minimizing both delay and transcoding errors. However, an all-digital process should at least eliminate nonlinear errors and noise which could arise in digital-to-analog-to-digital sampling rate conversions.

## REFERENCES

[1] T. Doi, Y. Tsuchiya, and A. Iga, "On Several Standards of Forms for Converting PCM Signals into Video Signals," Tech. Grp. of Magnetic Recording, TG MR 77-24 (1977-11) IECE Japan, Nov. 1977.

[2] K. Tanaka and Y. Ishida, "Sampling Frequency Consideration," *J. Audio Eng. Soc.*, vol. 26, pp. 248–250 (Apr. 1978).

[3] A. Heaslett, "Some Criteria for the Selection for Sampling Rates in Digital Audio Systems," *J. Audio Eng. Soc.*, vol. 26, pp. 66–70 (Jan./Feb. 1978).

[4] T. Doi, "On the Selection of the Higher Sampling Frequency," presented verbally to the AES meeting on digital audio in Los Angeles, Apr. 29, 1978 (to be published).

[5] J. J. Gibson, report presented verbally to the AES meeting on digital audio, Los Angeles, Apr. 29, 1978 (to be published).

[6] "Four-Channel PCM Processor Developed for Use with U-matic Cassette VTR," TEAC Corporation of America, P.O. Box 750, Montebello, CA 90640, TEAC Tech. Inform. 33204, Nov. 1977.

[7] H. Nakajima, T. Doi, Y. Tsuchiya, A. Iga, and I. Ajimine, "A New PCM Audio System as an Adapter of VTR," presented at the 60th Convention of the Audio Engineering Society, Los Angeles, May, 1978.

[8] M. Kosaka, K. Odaki, M. Tsuchiya, and R. Wada, "PCM Recording with Error-Correction Scheme," presented at the 60th Convention of the Audio Engineering Society, Los Angeles, May, 1978.

[9] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, vol. 65, pp. 1558–1564 (Nov. 1977).

[10] A. J. Jerri, "The Shannon Sampling Theorem, Its Various Extensions and Applications: A Tutorial Review," *Proc. IEEE*, vol. 65, pp. 1565–1596 (Nov. 1977).

[11] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE*, vol. 66, pp. 51–83 (Jan. 1978).

## BIBLIOGRAPHY

[12] Session E of the 60th Convention of the Audio Engineering Society, Los Angeles, May, 1978. Not all of these are in preprint form. See also the Briefs section of the *Journal* for other reports presented at the Los Angeles meeting on digital audio, April 29-30, 1978.

[13] D. A. Bell, *Information Theory and Its Engineering Applications*, (Pitman, London, 1972), chap. 2–6.

[14] W. H. Pierce, *Failure-Tolerant Computer Design* (Academic Press, New York, 3rd ed., 1965), chap. 7.