



Audio Engineering Society

Convention Paper

Presented at the 121st Convention
2006 October 5th-8th, San Francisco, CA, USA

This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

JITTER SIMULATION IN HIGH RESOLUTION DIGITAL AUDIO

M.O.J. Hawksford

University of Essex, Colchester, Essex, UK CO4 3SQ
mjh@essex.ac.uk

ABSTRACT

To reconstruct an audio waveform samples must be located precisely in time. Practical systems have sources of jitter described by both correlated and uncorrelated elements that result in low-level distortion. However, less well known is how different forms of jitter distort an audio signal. Jitter theory is developed to produce a simulator to enable jitter induced distortion to be determined. Distortion spectra can then be observed and time domain distortion auditioned. Jitter induced distortion is compared to a range of errors, including DAC errors and incorrect use of dither. System architectures studied include LPCM with up-sampling and noise shaping and SDM.

1. INTRODUCTION

This paper explores the area of jitter with a particular bias towards high resolution systems. Earlier work [1-8] has identified sources of jitter (which are well known) and also developed the mathematics which explains how jitter distorts a signal. Although jitter is not a fundamental distortion such as the errors derived from an incorrectly dithered quantizer [9], nevertheless it is pervasive in digital audio systems where to some degree all systems are defective in this area. Indeed in a high resolution system, distortion resulting from jitter can be more significant than quantizer distortion especially for systems using more than 20-bit resolution.

One of the major difficulties in quantifying and explaining the consequences of jitter is that there are many sources of jitter. Also, jitter can be classed into three basic forms (all can coexist) where there can be

periodic, correlated to audio and uncorrelated artifacts. Periodic jitter-related artifacts are further complicated as they can be linked, for example, to mains hum as well as the various clock signals present within equipment. Also, there can be correlated elements with the actual digital signals carrying the audio information. All these inter-related dependencies complicate the interpretation of jitter making it difficult for a simple jitter estimate or spectrum to be interpreted in terms of its subjective consequences.

As well as the numerous sources of error, the system architecture itself can influence the way jitter affects the resultant audio signal. For example, the use of noise shaping and up-sampling [10] with linear pulse code modulation (LPCM) alters the spectrum of the jitter induced distortion. Whilst, as suggested in an earlier paper [11], the use of a multiplying digital-to-analogue converter (DAC) with a raised cosine reference signal can in certain circumstances reduce distortion and

augment interpolation between samples prior to the low-pass filter reconstruction filter. There are also analogue amplifiers which when processing a sampled-data signal can produce distortion akin to correlated distortion [12]. Finally, the choice of 1-bit sigma-delta modulation (SDM) code [13], pulse-width modulation (PWM) code [14] or multi-level LPCM code [15] changes the nature of jitter distortion.

As well as presenting comparative discussion on these numerous system options/permutations the subject of jitter is approached from the perspective of simulation. A technique is discussed that shows how by applying the defining mathematics, the true jitter distortion can be extracted such that it can be auditioned to enable its sonic signature to be identified. The jitter simulator is defined so that it can operate at the native sampling rate of the system and also avoid problems of high frequency aliasing distortion that result from distortion products being generated “between samples”. The paper presents both a broad review of jitter mechanisms in relation to typical audio system formats and architectures and also describes the simulator in detail. Time domain and spectral examples are derived using an actual music sequence and 3-D spectral plots presented as a function of time and frequency to illustrate the relationship between signal and various forms of jitter.

This work is considered timely when seen in the broader context of high resolution audio systems. As such it demonstrates the importance of minimizing jitter if the true performance of a format is to be realized. It also seeks to identify whether there are significant performance differences when using various system architectures which consequently impact upon not only DAC systems but also on switching amplifiers using SDM signals. Finally, the paper attempts to make some system-level observations that may prove helpful to the designer especially those aspects interlinking digital with analogue circuitry.

2. DEFINITION OF JITTER

In most digital audio systems there is an expectation that time-sampling is uniform with samples separated by a constant time interval of $1/f_s$ where f_s is the sampling frequency. This applies for example, as to whether the system is LPCM, with or without noise shaping, and to SDM. However, in practical electronic systems there are numerous mechanisms that cause an undesirable time displacement, or *timing jitter* of each

sample. The consequence of jitter can be the introduction of modulation noise and distortion together with possible additional signal dependent distortion artifacts arising from non-linearity. To benchmark the level of timing jitter that can be of concern in high-resolution audio, consider the required timing accuracy to match the resolution of a 20-bit quantizer.

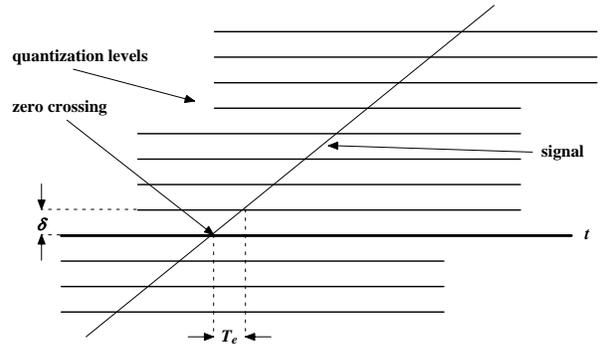


Figure 2-1 Distortion related to timing error.

A small segment of a sine wave is shown in Figure 2-1 that is to be sampled at a zero crossing where a timing error T_e is calculated such that the sample amplitude is in error by one quantum δ . To estimate T_e , assume a sine wave input $A \sin(\omega t)$ with amplitude A and angular frequency ω where the zero-crossing slope is,

$$\left. \frac{\partial \{A \sin(\omega t)\}}{\partial t} \right|_{\omega t=0} = A\omega \cos(\omega t)|_{\omega t=0} = A\omega$$

For an amplitude error δ then the corresponding timing error is,

$$T_e = \frac{\delta}{A\omega}$$

Hence, if \hat{A} is the maximum amplitude consistent with a M-bit uniform quantizer i.e. $\hat{A} = \delta 2^{(M-1)}$, then

$$T_e = \frac{2^{(1-M)}}{\omega} \dots 2-1$$

Equation 2-1 shows that the required level of jitter is an inverse function of frequency, where for example if $M = 20$ bit and $f = 15$ kHz, then $T_e = 20.2$ ps.

In considering the ramifications of jitter it is important to discriminate between jitter occurring at the analogue-to-digital converter (ADC) and jitter occurring at the DAC. Since uniform sampling is assumed, then ADC clock jitter creates distortion that is encoded into the digital signal and cannot be removed as there is no absolute reference to the actual sampling instants. However, jitter in the DAC clock is under the control of the system designer so means can be found to reduce its effect. Such techniques include precision phase-lock loops and the use of buffer memory to smooth out the signals being re-timed at the DAC [16].

In practical electronic systems there are many sources of jitter that can occur, these relate to noise in oscillators, signal reflections in digital inter-connections, power supply induced effects, mutual coupling between circuits and problems associated with non-optimum ground rails. Also, in networked audio substantial jitter occurs because of the indeterminate arrival of packet data. All these sources of jitter can be smoothed if some delay is acceptable, but in practice there is always a degree of residual jitter with elements that can be periodic together with random and correlated components with the digital data.

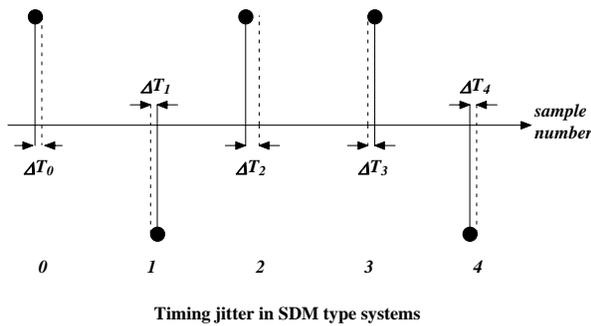


Figure 2-2 Type 1: Jitter error, samples of constant weight.

In this paper two classes of DAC topology are considered that have differences in response to jitter: Type 1 is defined where the reconstructed output samples have constant weight such as in SDM using switched-capacitor circuitry [17]. The basic Type 1 pulse structure is shown in Figure 2-2 where each pulse is assumed to have a constant area and where jitter causes only time displacements represented by ... $\Delta T_{r-1}, \Delta T_r, \Delta T_{r+1}, \dots$ etc.

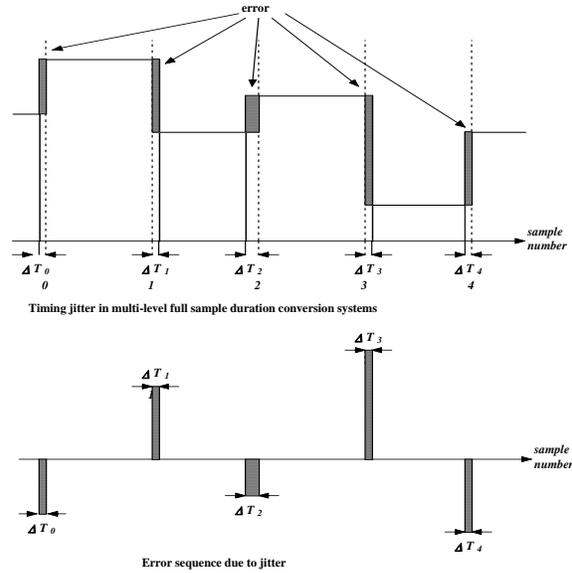


Figure 2-3 Type 2: Pulse-area error due to jitter with 100% sample reconstruction.

Type 2 jitter is defined where a DAC produces pulses of 100% sample duration. This second class is illustrated in Figure 2-3 where it can be seen that when adjacent pulses have differing amplitudes, jitter-induced pulse-edge modulation changes the sample area.

Type1: error (constant weight samples)

Consider sample r of a signal $y(r)$ that undergoes an instantaneous timing displacement ΔT_r . The corresponding error spectrum $E_r(f)$ then follows as,

$$\begin{aligned}
 E_r(f) &= \left\{ y(r) \left(1 - e^{-j2\pi f \Delta T_r} \right) \right\} e^{-j2\pi r f / f_s} \\
 &= \left\{ y(r) j2 e^{-j\pi f \Delta T_r} \left(\frac{e^{j\pi f \Delta T_r} - e^{-j\pi f \Delta T_r}}{j2} \right) \right\} e^{-j2\pi r f / f_s} \\
 &= \left\{ y(r) j2 e^{-j\pi f \Delta T_r} \sin(\pi f \Delta T_r) \right\} e^{-j2\pi r f / f_s}
 \end{aligned}$$

That is for $f \Delta T_r \ll 1$, then $E_r(f)$ approximates to,

$$E_r(f) \approx \left\{ j2\pi f y(r) \Delta T_r e^{-j\pi f \Delta T_r} \right\} e^{-j2\pi r f / f_s}$$

Hence, for a sequence of N samples the spectrum $E_N(f)$ follows as,

$$E_N(f) \approx j2\pi f \sum_{r=0}^{N-1} \left(y(r)\Delta T_r e^{-j2\pi f(0.5\Delta T_r + r/f_s)} \right) \dots 2-2$$

Type 2: error (100% duration samples)

Inspection of Figure 2-3 reveals that for the case of 100% duration pulses, the error resulting from jitter depends both upon the difference in sample amplitude between adjacent samples as well as the small jitter related time displacements of each sample. The resulting rectangular error pulses are shown in the lower section of Figure 2-3.

For sample r , the error due to jitter is a rectangular pulse of amplitude $y(r) - y(r-1)$ and width ΔT_r , where its centre is displaced by $0.5\Delta T_r$ from the optimum sampling position. Hence, the corresponding error spectrum $E_r(f)$ is,

$$E_r(f) = \left\{ [y(r) - y(r-1)]\Delta T_r e^{-j\pi f \Delta T_r} \text{sinc}(\pi f \Delta T_r) \right\} e^{-j2\pi r f / f_s}$$

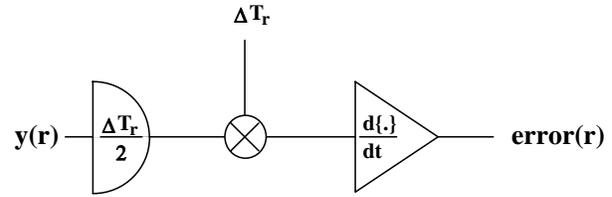
However, because for practical levels of jitter and for $f < f_s$, $f \Delta T_r \ll 1$ then $\text{sinc}(\pi f \Delta T_r) \approx 1$, whereby

$$E_r(f) \approx \left\{ [y(r) - y(r-1)]\Delta T_r e^{-j\pi f \Delta T_r} \right\} e^{-j2\pi r f / f_s}$$

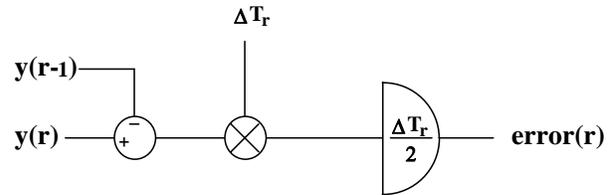
Hence, summing over N samples the composite error spectrum is,

$$E_N(f) \approx \sum_{r=0}^{N-1} \left([y(r) - y(r-1)]\Delta T_r e^{-j2\pi f(0.5\Delta T_r + r/f_s)} \right) \dots 2-3$$

Equations 2-2 and 2-3 reveal the well known trend in the error spectrum associated with jitter where for Type 1, the spectrum depends upon $y(r)\Delta T$ with an applied post 6 dB/octave spectral weighting while for Type 2, the post spectral weighting is independent of frequency but the error now depends upon the differential of the signal, i.e. $[y(r) - y(r-1)]\Delta T_r$. These two distortion mechanisms are shown in simplified form in Figure 2-4.



Type 1 jitter induced error.



Type 2 jitter induced error.

Figure 2-4 Simplified models of Type 1 and Type 2 error generation resulting from jitter.

3. SOURCES OF JITTER

There are numerous sources of jitter in digital audio systems and in practice it is this factor that leads to its complicated form. Some generic observations are as follows:

- The jitter sequence ΔT_r can have both a correlated and uncorrelated relationship to the audio signal as well as periodic elements.
- Uncorrelated jitter introduces noise where the noise amplitude is modulated either by the signal or the differential of the signal.
- Correlated and periodic jitter results in non-linear *intermodulation distortion* due either to $y(r)\Delta T_r$ or $\{y(r) - y(r-1)\}\Delta T_r$, together with spectral weightings described by Equations 2-2 and 2-3.
- Non-linearity in electronics can introduce similar effects to jitter [12] especially in the transresistance stage of a DAC when processing sampled data.

Uncorrelated noise results principally from thermal noise sources within electronics and causes phase modulation in both master and phase-lock loop

oscillator types used to determine sampling. Ultimately the level of phase noise is a function of oscillator bandwidth, so in order to produce very low noise levels crystal oscillator circuits are often the preferred choice.

Uncorrelated jitter, although it can result in modulation noise, is generally believed to be more benign compared to jitter that has a correlation with the audio data or has a relationship to a periodic signal such as mains hum. It has been shown [2] for example that when a digital audio signal is transmitted using a protocol such as SPDIF, then a limit in channel bandwidth results in jitter in the data transitions where the jitter pattern has a strong dependence on the audio data. This was demonstrated at the 93rd AES convention, where using a band-limited channel and deriving a signal from the control signal of a voltage-controlled oscillator (VCO) within a phase-lock loop (PLL), that the actual audio signal even though highly distorted, could be discerned. Of course such jitter contamination is not a fundamental impairment and can be filtered using appropriate PLL techniques possibly enhanced by a variable read-write buffer memory.

The success of jitter reduction techniques is often in the control of fine detail of circuit operation. For example, in a digital PLL [18] it is possible to design a loop filter with low bandwidth in order to reduce noise in the input clock. However, in some PLL systems it can sometimes be observed that when the input clock edge and the output VCO clock edge are almost coincident, that due to digital signal elements contaminating the ground rail or possibly because of small edge related “glitches” on the power supply rails, there can be a small range where the input edge directly “captures” the output edge directly, effectively bypassing the loop filters. Although this undesirable lock range is normally very small, if it exists, nevertheless it allows a sudden jump in output jitter level as for this false lock condition the PLL is bypassed. Again proper ground design and power supply decoupling can minimize the effect but it is a potential hazard.

Many jitter problems result directly from the numerous clocks that are distributed within digital audio systems. Logic gates inevitably produce transients on the power supply and also because of the resulting high bandwidth signals present in the ground rail, these can contaminate the associated sub-systems, this is why systems with multiple power supplies are often employed where a conceptual structure highlighting ground rail contamination is shown in Figure 3-1.

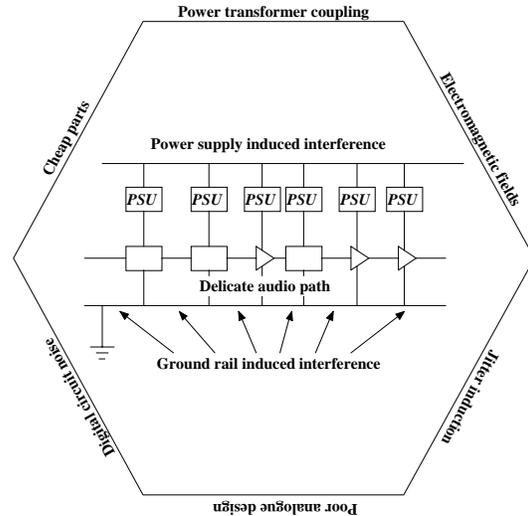
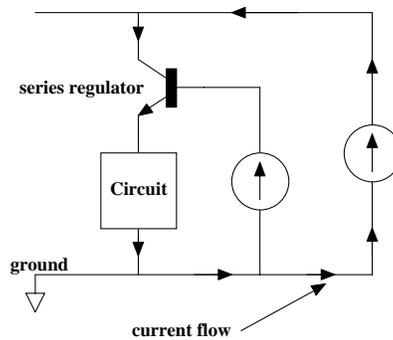
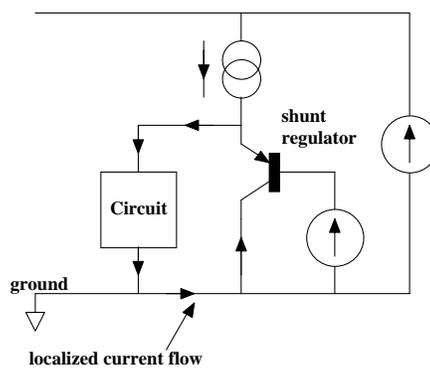


Figure 3-1 Conceptual cascaded processing stages showing ground rail contamination.



Series regulated power supply .



Shunt regulated power supply .

Figure 3-2 Series and shunt regulated power supply to show localization of current flow.

However, in such systems it is prudent to design circuits such that the high speed signal currents circulate only in local paths in order to prevent ground rail contamination. In Figure 3-2 this principle is illustrated by comparing a series regulated power supply with a shunt regulated power supply where the latter can localize ac current flow and thus protect the ground rail. Another circuit technique that has found application in multi-channel mixing consoles is the “dust bin”, a circuit designed to deflect signal currents away from the ground rail thus keeping it free of signal current enabling a better approximation to an equi-potential surface. This second circuit technique is shown in Figure 3-3.

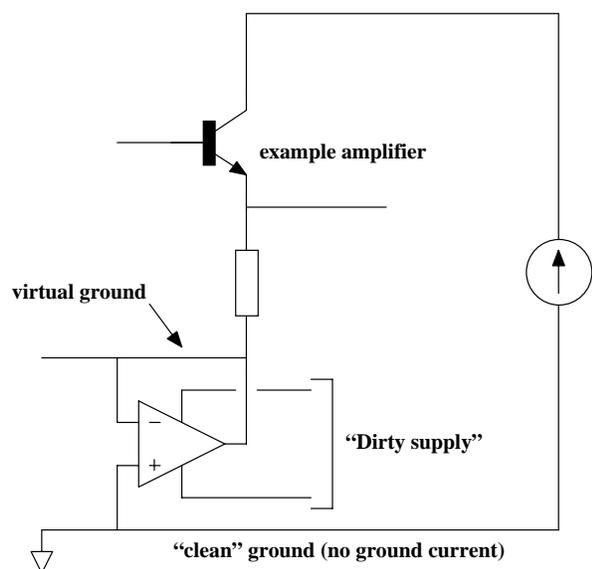


Figure 3-3 “Dustbin” used to eliminate current in ground rail and thus maintain equi-potential surface.

Yet a further type of jitter induction can occur in analogue circuits that have to process rapidly changing signals that are linked to a sampling clock. The most critical circuit is the transresistance amplifier or I/V stage located at the output of most multi-level DACs. This distortion mechanism has already been the subject of analysis in an earlier paper [12] so only the most salient features are highlighted. Consider the widely adopted transresistance amplifier shown in Figure 3-4 that uses a single operational amplifier in a shunt-feedback configuration. When combined with a current-output DAC this circuit has to amplify signals which at the sampling transitions exhibit rapid change. Consequently, any non-linearity in the amplifier will

cause the output signal of the amplifier to deviate from its ideal linear trajectory. In Figure 3-5 a gross example is shown where there is momentary slope overload; however even a modest non-linearity has a similar consequence although obviously the error is of a lower magnitude.

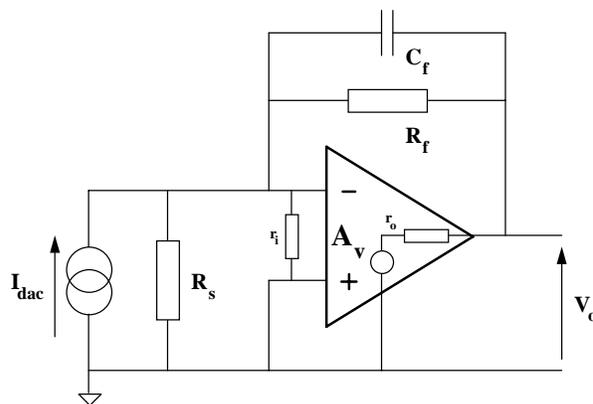


Figure 3-4 Transresistance amplifier formed using shunt-feedback operational amplifier.

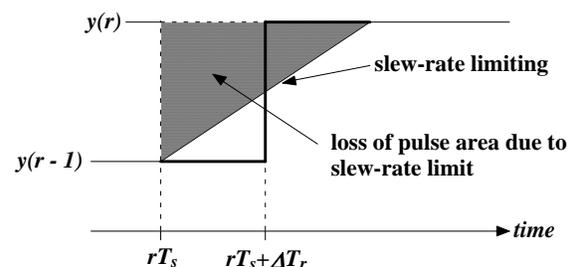


Figure 3-5 Illustration of the consequence of slew-rate limiting for a transresistance amplifier at a sample transition .

The effect of generating distortion such as a momentary slew-rate limit close to a sample boundary is that there is a small loss of pulse area where in effect an equivalent rectangular error pulse is displaced slightly in time as shown in Figure 3-5. Therefore it can be seen that this effect is very similar to that of jitter-induced distortion as illustrated in Figure 2-3 although in this case there is a high degree of correlation of error and the sampled audio signal.

In this class of problem it is instructive to calculate the change in differential time delay (akin to jitter) as a

function of changes in operational amplifier dc gain A_0 and dominant pole break frequency f_0 as these may be dynamically modulated under transient conditions, such as at a sample boundary. For the transresistance amplifier as illustrated in Figure 3-4 with feedback impedance Z_f , let the input current from the DAC be I_{dac} and the output voltage be V_0 . Assuming an ideal operational amplifier the target transresistance $Z_T(f)$ is defined,

$$Z_T(f) = \frac{V_0}{I_{dac}} = -Z_f \quad \dots 3-1$$

while for a practical operational amplifier, the actual transimpedance is $Z_A(f)$ can be expressed in terms of the virtual-earth input impedance z_{in} , where

$$V_0 = -I_{dac}Z_f + v_{in} = -I_{dac}Z_f + I_{dac}z_{in}$$

that is,

$$Z_A(f) = \frac{V_0}{I_{dac}} = -Z_f + z_{in} \quad \dots 3-2$$

Let the operational amplifier open-loop transfer function A_v be modeled in terms of its dc gain A_{v0} and n poles f_0, f_1, \dots, f_{n-1} , where

$$A_v = \frac{A_{v0}}{\sum_{r=0}^{n-1} \left(1 + j \frac{f}{f_r}\right)} \quad \dots 3-3$$

From classical feedback theory, the input impedance of the transimpedance amplifier is,

$$z_{in} = \frac{Z_f}{1 + A_v} \quad \dots 3-4$$

Hence,

$$Z_A(f) = -Z_f + \frac{Z_f}{1 + A_v} = -Z_f \frac{A_v}{1 + A_v} \quad \dots 3-5$$

The differential phase error $\Delta\phi$ which represents the phase difference between $Z_T(f)$ and $Z_A(f)$ is given by,

$$\Delta\phi = \text{phase}\left(\frac{Z_A(f)}{Z_T(f)}\right) = \text{phase}\left(\frac{A_v}{1 + A_v}\right) \quad \dots 3-6$$

whereby substituting for A_v from Equation 3-3,

$$\Delta\phi = -\text{phase}\left(1 + \frac{1}{A_{v0}} \sum_{r=0}^{n-1} \left(1 + j \frac{f}{f_r}\right)\right) \quad \dots 3-7$$

Now group delay T_{diff} as a function of phase is defined,

$$T_{diff} = -\frac{1}{2\pi} \frac{\partial \Delta\phi}{\partial f} \quad \dots 3-8$$

Hence, for a first-order amplifier where $n = 1$ then,

$$\Delta\phi = -\text{phase}\left(1 + \frac{1}{A_{v0}} \left(1 + j \frac{f}{f_0}\right)\right) \quad \dots 3-9$$

and noting for $\tan(\Delta\phi)|_{f \ll A_{v0}f_0} \approx -\left(\frac{f}{A_{v0}f_0}\right)$, then

$$T_{diff} \approx \frac{1}{2\pi A_{v0}f_0} \quad \dots 3-10$$

Consider the change in group delay ΔT_{diff} as a function of ΔA_{v0} and ΔA_0 , i.e.

$$\Delta T_{diff} = \frac{\partial T_{diff}}{\partial A_{v0}} \Delta A_{v0} + \frac{\partial T_{diff}}{\partial f_0} \Delta f_0 \quad \dots 3-11$$

whereby,

$$\Delta T_{diff} = \frac{-1}{2\pi A_{v0}f_0} \left(\frac{\Delta A_{v0}}{A_{v0}} + \frac{\Delta f_0}{f_0}\right) \text{ second} \quad \dots 3-12$$

As an example, let $f_0 = 100 \text{ Hz}$ and $A_{v0} = 10^5$, where for a 1% gain change, the timing error (i.e. differential group delay) follows from Equation 3-12 as about 159 ps.

4. JITTER-INDUCED DISTORTION MODELLING

In this Section the method of distortion generation as a function of a sampled signal and a predefined jitter sequence is described. This allows the distortion to be analyzed spectrally and for an audio file to be produced so that the actual distortion can be auditioned. It is assumed that the sampled source data is a LPCM file with a sampling rate f_s Hz and bit depth M , where by using oversampling and noise shaping it is then possible to adapt the technique to a variety of systems. It is also assumed that the jitter sequence is presented in a similar sampled data format so that it can be arbitrarily defined such that either correlated or uncorrelated sequences can be formed or indeed, sequences that are periodic and synchronized for example to mains hum. Hence, each jitter sample specifies the timing error that a corresponding audio sample will experience.

Another requirement of the simulator is that distortion due to the non-linear interaction of jitter and audio data will be calculated at the same sampling rate f_s and not be contaminated by any form of aliasing distortion that is not representative of the distortion process. This allows the distortion file to be auditioned directly, or if required, the file can be imported into an audio workstation to change sampling rate and/or bit depth to be compatible with a specific reproduction system.

Consider the operation of a real-world audio DAC where the clock is contaminated with jitter. The output signal will consist of the desired audio (represented by the LPCM source sequence) together with in-band distortion as a result of intermodulation between audio and jitter. However, there will also be high frequency components which are a combination of the normal spectral replication due to sampling but which are further modified due to the time modulation of jitter. Observe that in this case the samples are no longer quite equally spaced because of the presence of timing jitter.

In order to determine the distortion it is necessary to recover both the undistorted sequence and the distorted sequence, calculate the difference signal and then bandlimit this error signal to $f_s/2$ Hz so that it can be correctly recorded in LPCM. Ideally this should be achieved without recourse to oversampling thus allowing only signal processing to be performed at f_s Hz.

Hence to summarize the requirements:

- Jitter simulation must process sampled data at f_s Hz.
- Simulation must output a uniformly sampled sequence that contains the error resulting from timing jitter when both the audio signal and the jitter signal are specified as sampled data sequences.
- Since up-sampling at the point of measurement is not employed, care must be taken to prevent aliasing distortion arising from ultrasonic signal components formed by sample jitter, noting the jitter distortion is effectively re-sampled.

Simulation should model both Type 1 and Type 2 systems as defined in Section 2. i.e.

- Type 1: Time displaced pulses with constant area.
- Type 2: Time displaced samples where a sample-and-hold function creates 100% pulses. Hence, simulation must take account of timing errors and, for the 100% pulse case, changes in pulse area.

In order to bandlimit precisely the jittered data sequence and to emulate the reconstruction of an analogue signal, the audio sequence is convolved with a $\text{sinc}(x)$ function selected to have its zeros spaced at the nominal sampling interval of $T_s = 1/f_s$. Hence for sample r that experiences instantaneous jitter ΔT_r , the $\text{sinc}(x)$ function is centered at time $rT_s + \Delta T_r$ giving,

$$\text{sinc}(x) = \frac{\sin(\pi(f_s(t - \Delta T_r) - r))}{\pi(f_s(t - \Delta T_r) - r)} \quad \dots 4-1$$

Consequently, for sample r with amplitude $y(r)$ and jitter ΔT_r the weighted and bandlimited $\text{sinc}(x)$ function $y_b(t)$ evaluated at time t becomes,

$$y(r)|_{\text{bandlimit}} \Rightarrow y_b(t) = y(r) \frac{\sin(\pi(f_s(t - \Delta T_r) - r))}{\pi(f_s(t - \Delta T_r) - r)} \quad \dots 4-2$$

Although Equation 4-2 describes the bandlimited sample at any time t , the simulator re-samples this function only at discrete times $t_m = mT_s$, thus $y_b(mT_s)$ becomes,

$$y_b(mT_s) = y(r) \frac{\sin(\pi(r - m + f_s \Delta T_r))}{\pi((r - m) + f_s \Delta T_r)} \dots 4-3$$

The evaluation of $y_b(mT_s)$ can be simplified to reduce computation time by observing that the level of jitter normally encountered in a digital audio system is only a small fraction of the sampling period T_s , that is $f_s \Delta T_r \ll 1$. Hence, from Equation 4-3,

$$y_b(mT_s) = y(r) \frac{\sin(\pi(r - m)) \cos(\pi f_s \Delta T_r) + \cos(\pi(r - m)) \sin(\pi f_s \Delta T_r)}{\pi((r - m) + f_s \Delta T_r)}$$

which reduces to,

$$y_b(mT_s) = (-1)^{r-m} y(r) \frac{\sin(\pi f_s \Delta T_r)}{\pi((r - m) + f_s \Delta T_r)} \dots 4-4$$

Noting $f_s \Delta T_r \ll 1$ implies each $\text{sinc}(x)$ function associated with sample r is sampled very close to its zero crossings, i.e. $\sin(\pi f_s \Delta T_r) \rightarrow \pi f_s \Delta T_r$, then the re-sampled bandlimited response $y_b(mT_s)$ of Equation 4-4 can be approximated to:

For $r - m = 0$

$$y_b(mT_s)|_{r-m=0} = y(r) \frac{\sin(\pi f_s \Delta T_r)}{\pi f_s \Delta T_r} \approx y(r) \dots 4-5a$$

otherwise, for $r - m \neq 0$

$$y_b(mT_s)|_{r-m \neq 0} = (-1)^{r-m} y(r) \frac{\sin(\pi f_s \Delta T_r)}{\pi((r - m) + f_s \Delta T_r)} \approx (-1)^{r-m} y(r) \frac{\{f_s \Delta T_r\}}{r - m} \dots 4-5b$$

Hence using the approximation described by Equations 4-5a and 4-5b, the sampled bandlimited error response associated with sample r takes the odd symmetric vector form,

$$y(r) \begin{bmatrix} \dots & \frac{+\{f_s \Delta T_r\}}{3} & \frac{-\{f_s \Delta T_r\}}{2} & \frac{+\{f_s \Delta T_r\}}{1} & 1 \\ & \frac{-\{f_s \Delta T_r\}}{1} & \frac{+\{f_s \Delta T_r\}}{2} & \frac{-\{f_s \Delta T_r\}}{3} & \dots \end{bmatrix}$$

Using this simplified vector for convolution, the sampled error for Type 1 and Type 2 systems can be evaluated:

Type 1 case

For Type 1 jitter, the bandlimited sampled error sequence $J_{err1}(m, r)$ evaluated at sample m and associated with sample r can be calculated from Equation 4-4 as the difference between the jittered sequence and the non-jittered sequence, that is

$$J_{err1}(m, r) = (-1)^{r-m} y(r) \left\{ \frac{\sin(\pi f_s \Delta T_r)}{\pi((r - m) + f_s \Delta T_r)} - \frac{\sin(0)}{\pi(r - m)} \right\} \dots 4-6$$

However, the second term in Equation 4-6 is always zero except for $r - m = 0$ where it is 1, hence when the simplification described by Equations 4-5a and 4-5b is applied, it can be seen to be identical to setting the central term in the convolution vector to zero, that is Equation 4-6 approximates to,

$$y(r) \{f_s \Delta T_r\} \left[\dots \frac{+1}{3} \quad \frac{-1}{2} \quad \frac{+1}{1} \quad 0 \quad \frac{-1}{1} \quad \frac{+1}{2} \quad \frac{-1}{3} \dots \right]$$

The simulation for Type 1 jitter can now be seen as calculating a sequence of error samples for each sample r and using the asymmetric vector in convolution,

$$\left[\dots \frac{+1}{3} \quad \frac{-1}{2} \quad \frac{+1}{1} \quad 0 \quad \frac{-1}{1} \quad \frac{+1}{2} \quad \frac{-1}{3} \dots \right]$$

which for sample r is weighted by $y(r) \{f_s \Delta T_r\}$. It should be observed that because the level of jitter is normally small compared to a sample period then the vector describing the re-sampled time dispersion due to low-pass brickwall filtering does not change its shape with ΔT_r , there is only overall amplitude weighting of the sequence. From a computational perspective this

approximation represents a useful speed advantage as sample-specific vectors do not require calculation. The convolution is completed by summing all the contributions over the range of R samples to form the total Type 1 error $JT_{err1}(m)$, where

$$JT_{err1}(m) = \sum_{r=0}^{R-1} J_{err1}(m, r) \quad \dots 4-7$$

However, in practice it is necessary to truncate the series approximation to $\text{sinc}(x)$ used to low-pass filter the error in order to form a finite impulse response (FIR) filter to facilitate convolution where the consequences of truncation are discussed in Section 5.

Type 2 case

The process to determine the error for the case of Type 2 error follows a similar procedure to that described for Type 1 jitter. However, as discussed in Section 2 and illustrated in Figure 2-3, the full bandwidth error resulting from jitter for sample r is now a rectangular pulse of amplitude $y(r) - y(r-1)$ and of duration ΔT_r , where the simulation process including filtering and re-sampling is shown in Figure 4-1.

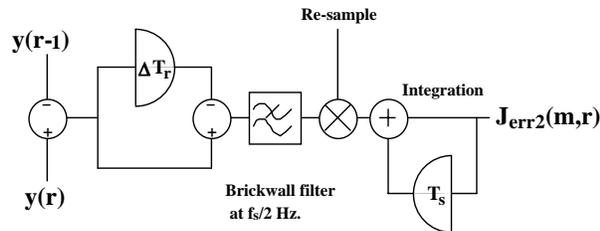


Figure 4-1 Type 2 jitter simulator incorporating sinc(x) low-pass filtering and integration to construct rectangular error pulses.

As shown, the derived jitter error associated with sample r is formed by subtracting the inter-sample difference signal $y(r) - y(r-1)$ (assumed to have an impulsive form) from a delayed version of itself, which ensures a zero mean and where the delay is set equal to the jitter associated with sample r ; thus by integrating the two time displaced samples of opposite polarity, a rectangular error pulse of width ΔT_r is formed. However, the error pulse requires low-pass filtering to one half the sampling frequency prior to re-sampling where this is performed by filtering and then re-

sampling the two impulsive sequences to yield time sampled $\text{sinc}(x)$ functions. Adapting Equation 4-6, the Type 2 jitter vector $J'_{err2}(m, r)$ then follows as,

$$J'_{err2}(m, r) = (-1)^{r-m} \{y(r) - y(r-1)\} \left\{ \frac{\sin(\pi f_s \Delta T_r)}{\pi((r-m) + f_s \Delta T_r)} - \frac{\sin(0)}{\pi(r-m)} \right\} \quad \dots 4-8$$

Because $f_s \Delta T_r \ll 1$, an approximation vector similar to the Type 1 case is then derived from Equation 4-8 as,

$$\{y(r) - y(r-1)\} \{f_s \Delta T_r\} \left[\dots \frac{+1}{3} \quad \frac{-1}{2} \quad \frac{+1}{1} \quad 0 \quad \frac{-1}{1} \quad \frac{+1}{2} \quad \frac{-1}{3} \dots \right]$$

However, because integration is required to form a rectangular error pulse, then as shown in Figure 4-1, integration must also be applied to the bandlimited vector. Hence, the uniformly sampled sequence $J'_{err2}(m, r)$ is integrated to form the actual error sequence $J_{err2}(m, r)$. This integration is formed in the discrete domain by a running accumulator such that,

$$J_{err2}(m, r) = \sum_{s=0}^{r-1} J'_{err2}(s, r) \quad \dots 4-9$$

To determine the final error $JT_{err2}(m)$ at sample m including the contributions from all samples for an audio sequence of R samples, a similar summation procedure to that described in Equation 4-7 is followed, whereby

$$JT_{err2}(m) = \sum_{r=0}^{R-1} J_{err2}(m, r) \quad \dots 4-10$$

5. SELECTION OF FIR FILTER LENGTH

In this Section the jitter simulation process is applied to both a Type 1 and Type 2 system where both the audio and the jitter sequences are uncorrelated white noise with triangular probability distribution functions (PDF). The objective is to confirm the form of the resulting distortion spectrum and to investigate the effect of

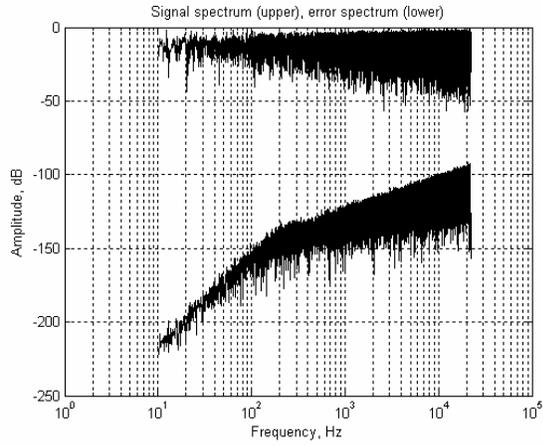


Figure 5-1a Jitter FIR test Type 1 for $R = 2^8$.

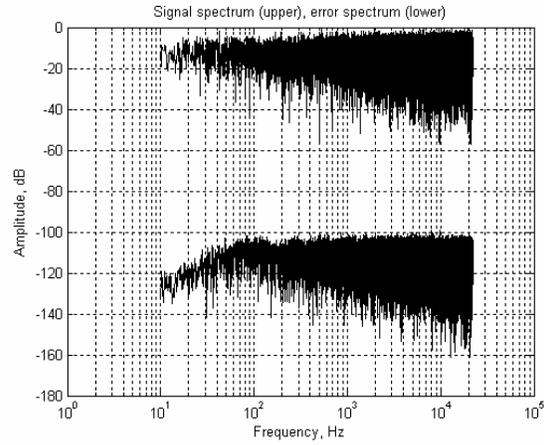


Figure 5-2a Jitter FIR test Type 2 for $R = 2^8$.

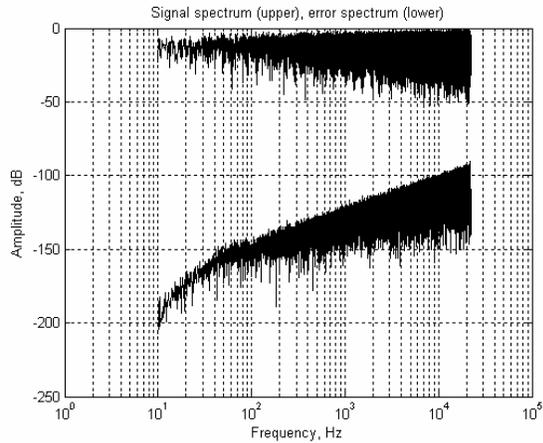


Figure 5-1b Jitter FIR test Type 1 for $R = 2^{10}$.

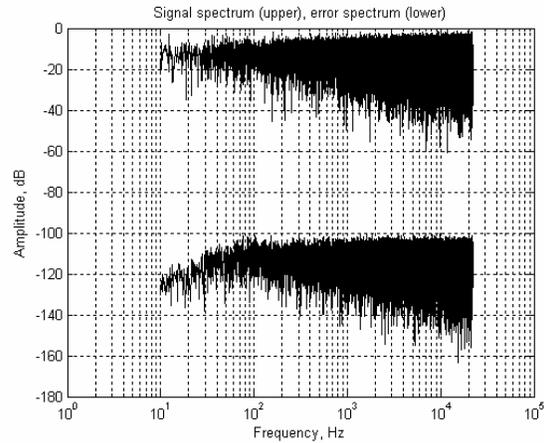


Figure 5-2b Jitter FIR test Type 2 for $R = 2^{10}$.

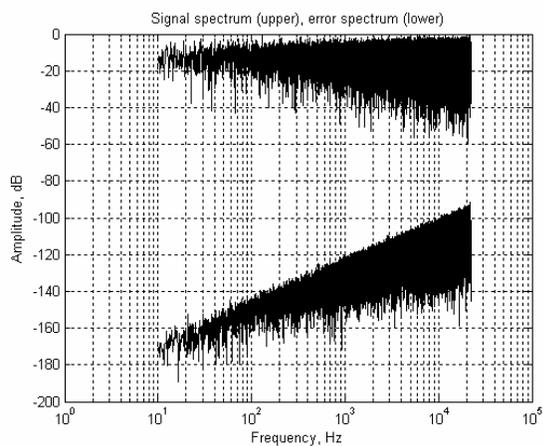


Figure 5-1c Jitter FIR test Type 1 for $R = 2^{12}$.

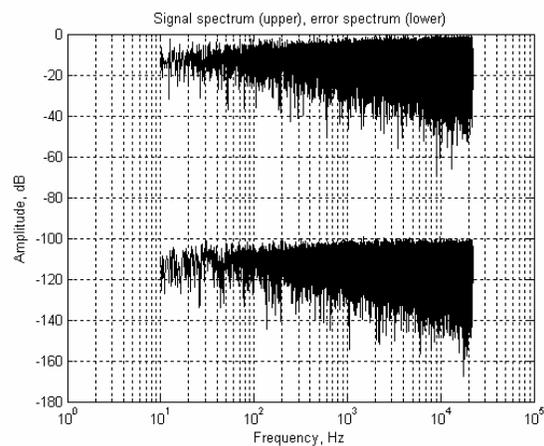


Figure 5-2c Jitter FIR test Type 2 for $R = 2^{12}$.

changing the length R of the FIR filter used to bandlimit the jitter spectrum prior to re-sampling. Word depth for the audio sequence was set at 24 bit. Spectra were computed over 2^{18} samples with peak-to-peak jitter level selected to be 1 ns. For each jitter type three FIR filter lengths R were selected of $2^8, 2^{10}$ and 2^{12} , where for Type 1 the respective results are shown in Figure 5-1(a,b,c) and for Type 2 in Figure 5-2(a,b,c).

As anticipated for $R=2^{12}$, the spectrum for Type 1 shows the distortion spectrum with a 6-dB/octave slope (see Equation 2-2) while for Type 2 the spectrum is approximately constant with frequency (see Equation 2-3). The advantage in Type 1 of the pulse area not being modulated by jitter is evident by the significantly lower spectral energy at low frequency. However, as the FIR filter length is reduced both sets of spectra show deviation at lower frequencies from the theoretical becoming progressively more distorted as the length is reduced. From these observations it is concluded that a length of $R=2^{12}$ yields satisfactory results and this value is employed in all subsequent examples.

6. LPCM JITTER SIMULATION

To explore the effect of uncorrelated and periodic jitter in LPCM four sets of simulations were performed using a combination of a 44.1 kHz @ 16 bit sampled audio signal and specific jitter sequences all computed over 2^{18} samples using a generating function normalized to 1 ns, that for sample r has the generic form,

$$m(r) = 10^{-9} \left\{ \beta_n rd + \sum_{p=1}^3 \alpha_p \sin(2\pi r f_p / f_s) \right\} ns \quad \dots 6-1$$

where α_p is the p^{th} sinewave amplitude and β_n the amplitude weighing of a random number rd taken from a noise vector having a rectangular PDF spanning an amplitude range -1 to 1. The three selected jitter sequences (Examples 1 to 3) are defined as follows:

Example 1

$$\begin{matrix} \beta_n = 1 \\ f_1 = 0 & f_2 = 0 & f_3 = 0 \\ \alpha_1 = 0 & \alpha_2 = 0 & \alpha_3 = 0 \end{matrix}$$

Example 2

$$\begin{matrix} \beta_n = 0 \\ f_1 = 44100+50 & f_2 = 44100-50 & f_3 = 0 \\ \alpha_1 = 1 & \alpha_2 = -1 & \alpha_3 = 0 \end{matrix}$$

Example 3

$$\begin{matrix} \beta_n = 0 \\ f_1 = 50 & f_2 = 100 & f_3 = 150 \\ \alpha_1 = 1 & \alpha_2 = 0.5 & \alpha_3 = 0.25 \end{matrix}$$

Example 4 Miller Test with Examples 1 to 3.

The Miller test specified in example 4 is widely used for jitter assessment [19]. The LPCM audio signal used for a Miller test consists of two superimposed periodic sequences where the first is a high level signal having consecutive sample values,

$$\dots 0.5 \quad 0.5 \quad -0.5 \quad -0.5 \quad 0.5 \quad 0.5 \quad -0.5 \quad -0.5 \quad \dots$$

The second sequence is a low-level 1:1 mark-space ratio square wave with a period of 192 samples and an amplitude equivalent to the least significant bit of a 16-bit word as the least significant bit is toggled every 96 samples. Hence,

$$\text{High frequency sequence is } 44100/4 = 11.025 \text{ kHz}$$

$$\text{Low frequency sequence is } 44100/192 = 229.6875 \text{ Hz}$$

There is no dither thus the sequence is noiseless. By observing the resulting distortion spectrum around 11.025 kHz, sidebands reveal clues both to the level of jitter as well as aspects of its formation. The LPCM system for Type 2 jitter (representative of multi-level LPCM) was investigated using a short extract of music where for Examples 1 to 3 results are presented in both time and frequency domains. Figures 6-1(a,b) to 6-3(a,b) show example specific time domain responses of music excitation and resulting distortion, while in Figures 6-1c to 6-3c corresponding 3-dimensional time-frequency plots are presented where the frequency analysis is split into 25 ms raised-cosine weighted blocks using the same computation procedure as presented in an earlier paper [20]. The time and frequencies axes are both linear. The results for each jitter example using the Miller test (where the Miller test sequence replaces the music sequence) are shown in Figures 6-4 to 6-6 for a Type 2 LPCM system sampled at 44.1 kHz. For these calculations the three jitter sequences of Examples 1 to 3 were used; however only spectra are shown zoomed to the range 10.4 kHz to 11.6 kHz in order to reveal the intermodulation components.

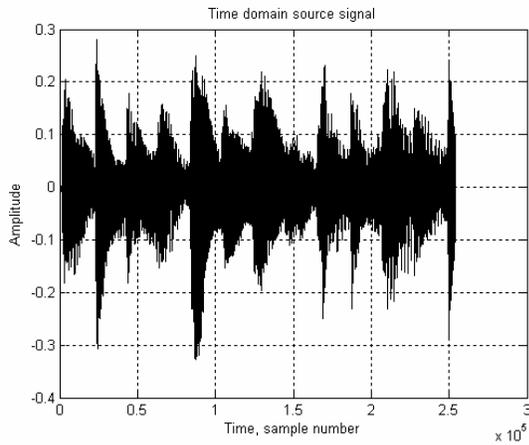


Figure 6-1a Example 1: Time domain music signal.

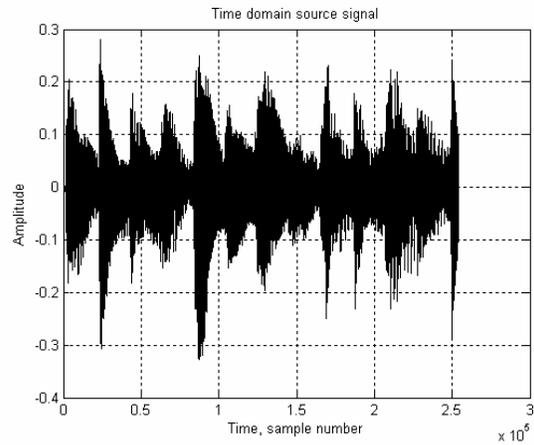


Figure 6-2a Example 2: Time domain music signal.

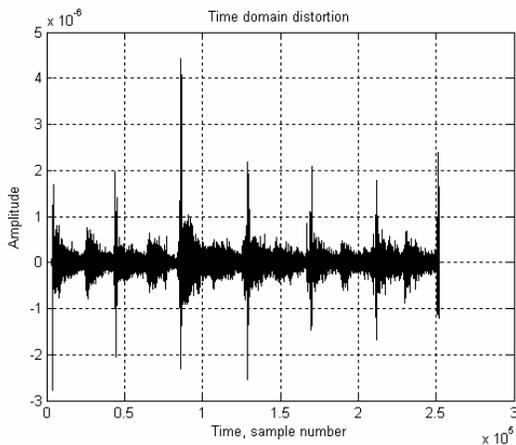


Figure 6-1b Example 1: Time domain distortion.

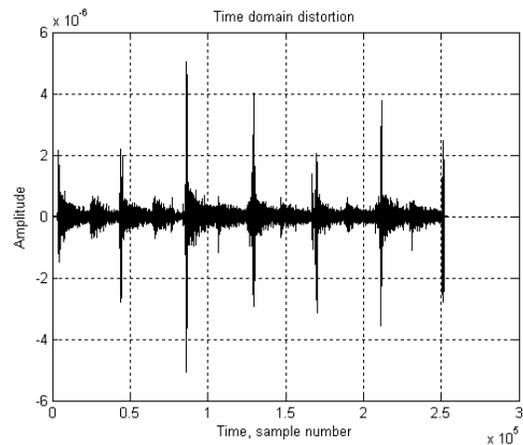


Figure 6-2b Example 2: Time domain distortion.

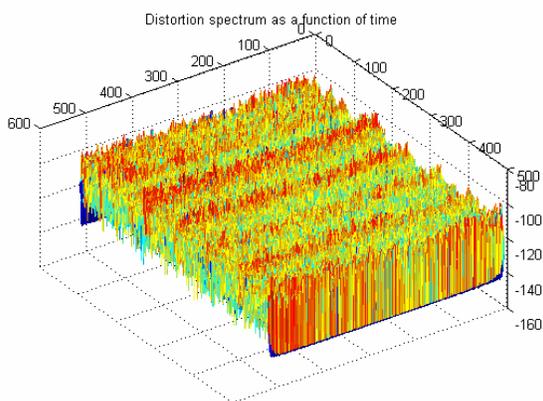


Figure 6-1c Example 1: 3-D time-spectral distortion.

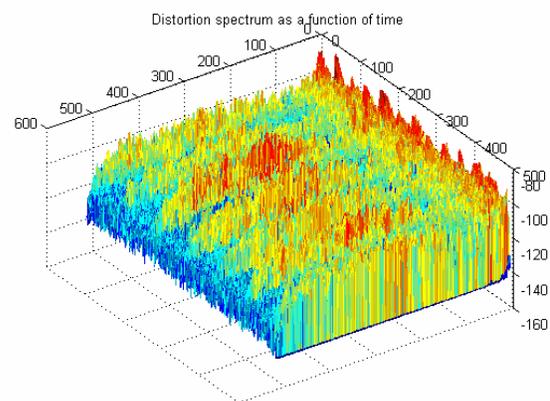


Figure 6-2c Example 2: 3-D time-spectral distortion.

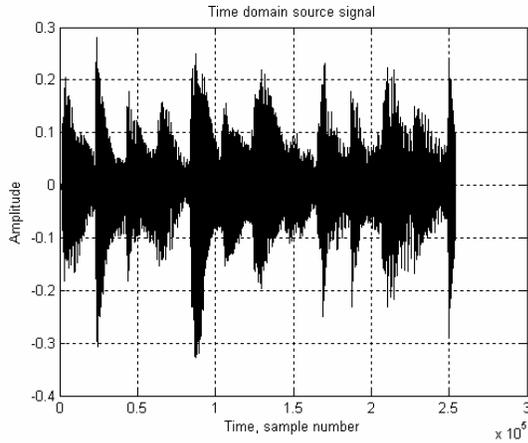


Figure 6-3a Example 3: Time domain music signal.

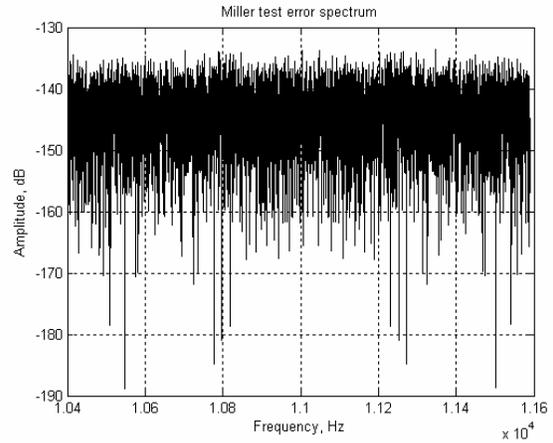


Figure 6-4 Example 1: Frequency domain Miller.

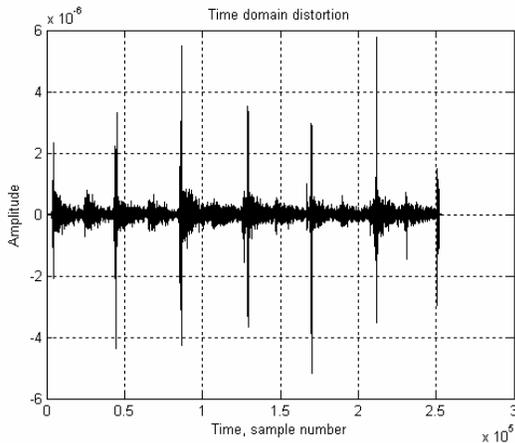


Figure 6-3b Example 3: Time domain distortion.

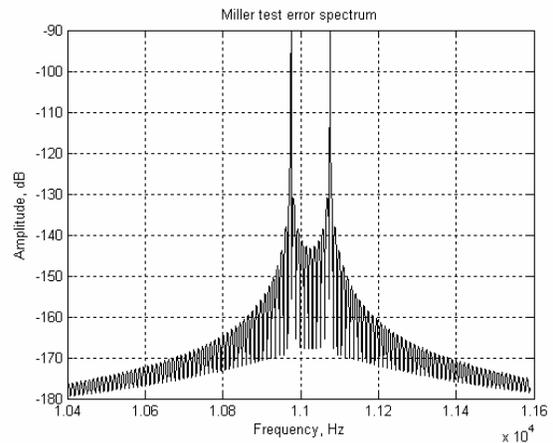


Figure 6-5 Example 2: Frequency domain Miller.

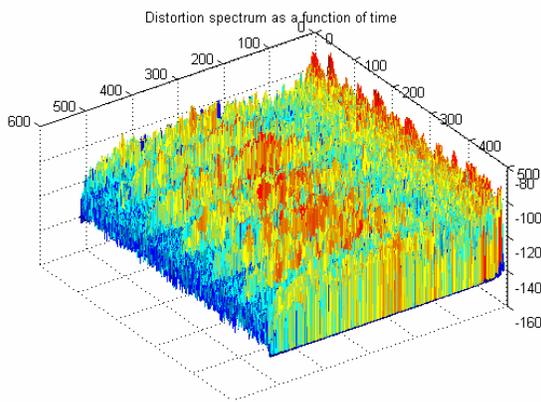


Figure 6-3c Example 3: 3-D time-spectral distortion.

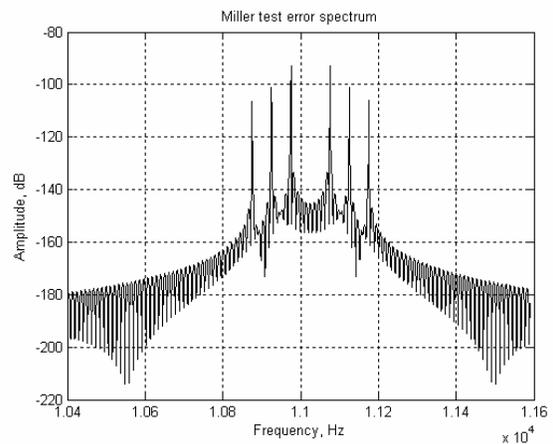


Figure 6-6 Example 3: Frequency domain Miller.

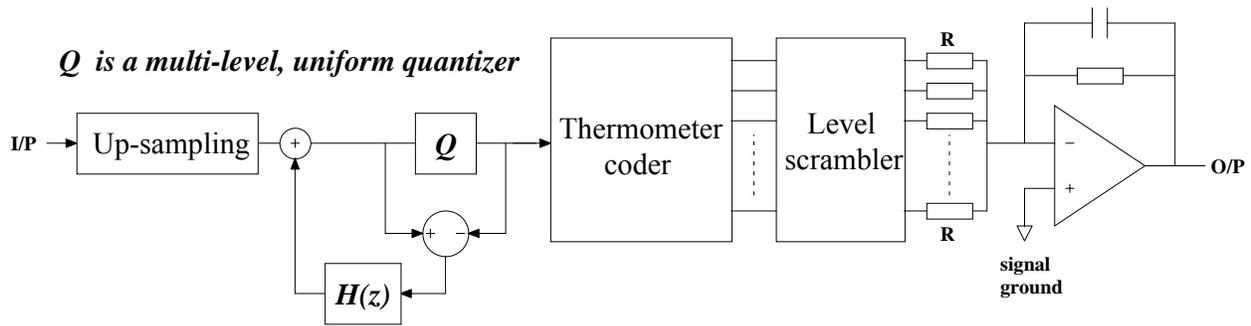


Figure 7-0 DAC structure with noise shaping.

7. JITTER IN UP-SAMPLED AND NOISE-SHAPED LPCM SYSTEMS

In this Section the simulator is extended to include oversampling and noise shaping, the latter allowing an exchange between sampling rate and DAC amplitude resolution. The objective is to explore the interaction of jitter with high-frequency quantization distortion and the consequence of using a higher sampling rate. A typical DAC architecture is shown in Figure 7-0 where the input audio signal is up-sampled from f_s Hz to f_{sam} Hz and the amplitude resolution reduced to simplify the digital-to-analogue converter stage. In this structure it is common for a scrambler [21] to be used to decorrelate distortion associated with systematic errors, however in the simulations presented here quantization level reconstruction is assumed perfect since the aim is to isolate distortion due only to the interaction of jitter with the audio data. The simulator was adapted to include sampling rate conversion based upon an integer, power-of-2 factor designated *over*, where

$$over = \frac{f_{sam}}{f_s} \quad \dots 7-1$$

Two dithered noise shaper options were included: second order [10] and Sony-FF SDM [22] (shown in Figure 8-1) but with quantization relaxed to include multi-level with an up-sampling conversion ratio of *over*=64. Other than the inclusion of these two additional processes, the simulator was in all other respects identical to the structure described in Section 4 and was configured both for Type 1 and Type 2 jitter. A *Matlab* script is presented in the Appendix, Section 12. The only limit imposed by computation is because of the high oversampling ratio, it became necessary to reduce the length of the audio sequence in order to keep the total number of processed elements constant.

In this Section all illustrative computations used a vector length of 2^{20} samples and a music extract with source parameter of 44.1 kHz @ 16 bit. Each set of results include time domain distortion, overall spectra of audio signal and distortion presented on a common graph to emphasize their relative level, together with a 3-D time-frequency plot to show spectral variation with time, again based upon the 25 ms block transform used in Section 6. Both Type 1 and Type 2 jitter models were simulated where the presentation of the results shows Type 1 to the left and Type 2 to the right. Because the comparative sets of spectra of signal and distortion (i.e. non 3-D plots) were computed for the total vector, they are averaged results and therefore do not reveal how distortion varies with time, consequently plots of time domain distortion are also shown.

The following simulations were performed where in each case *over*=64, jitter was scaled to a peak level of 1 ns and the output DAC incorporated multi-level LPCM:

- Figure 7-1: Jitter → Example 1, no noise shaping, output DAC resolution 16 bit.
- Figure 7-2: Jitter → Example 3, no noise shaping, output DAC resolution 16 bit.
- Figure 7-3: Jitter → Example 1, 2nd-order noise shaping, output DAC resolution 10 bit.
- Figure 7-4: Jitter → Example 3, 2nd-order noise shaping, output DAC resolution 10 bit.
- Figure 7-5: Jitter → Example 1, Sony-FF noise shaping, output DAC resolution 6 bit.
- Figure 7-6: Jitter → Example 3, Sony-FF noise shaping, output DAC resolution 6 bit.

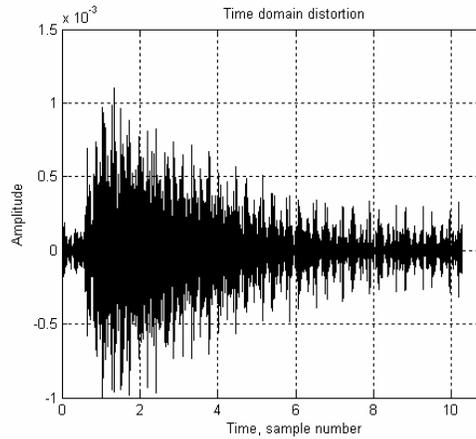


Figure 7-1a Example 1 with Type 1 jitter, no noise shaping: Time domain distortion.

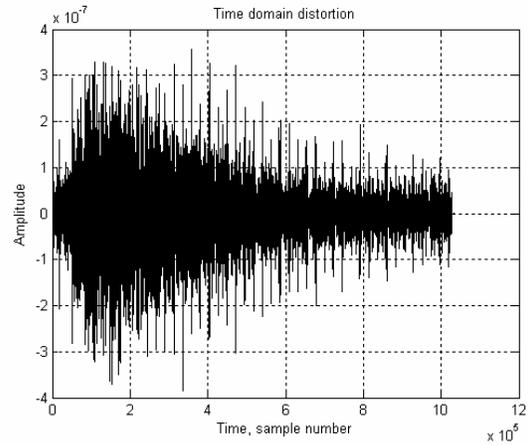


Figure 7-1d Example 1 with Type 2 jitter, no noise shaping: Time domain distortion.

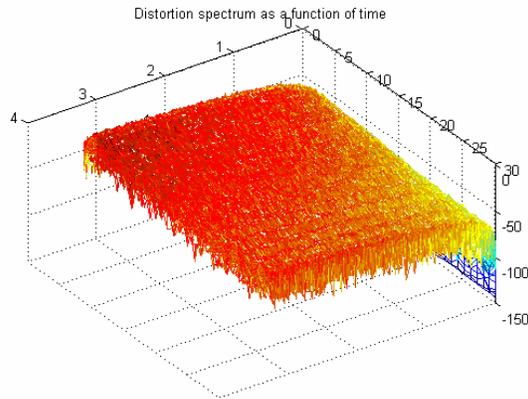


Figure 7-1b Example 1 with Type 1 jitter, no noise shaping: Time-frequency spectrum.

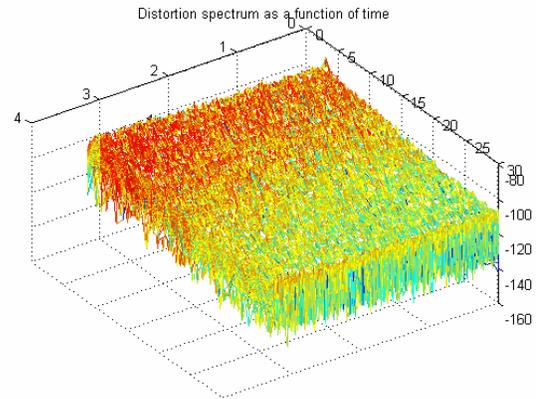


Figure 7-1e Example 1 with Type 2 jitter, no noise shaping: Time-frequency spectrum.

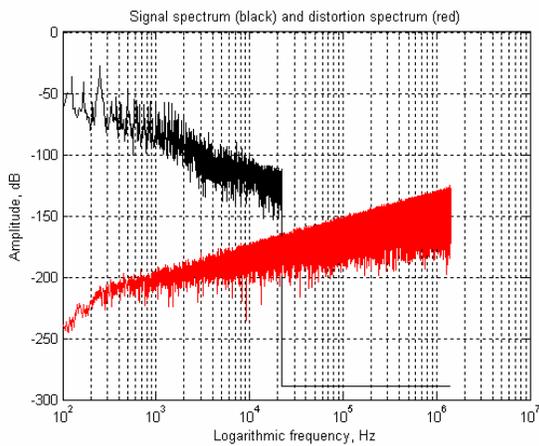


Figure 7-1c Example 1, Type 1 jitter, no noise shaping: Signal and distortion spectra.

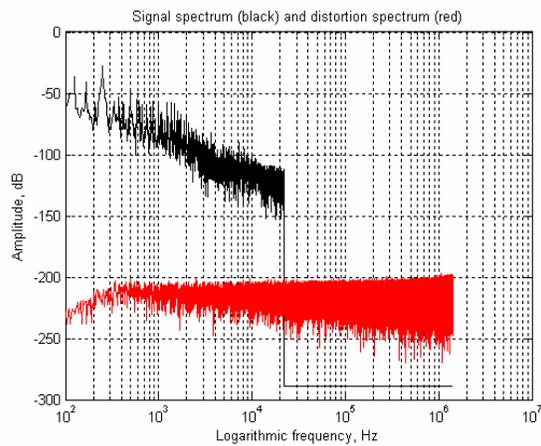


Figure 7-1f Example 1, Type 2 jitter, no noise shaping: Signal and distortion spectra.

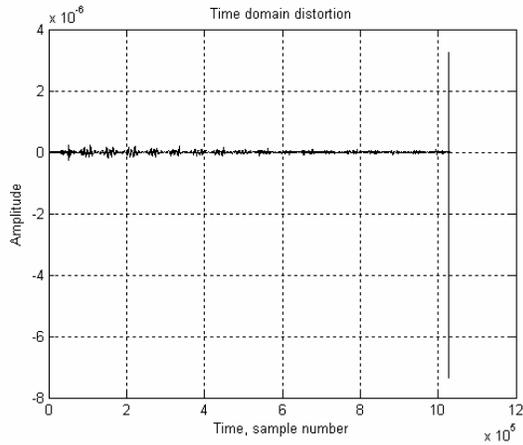


Figure 7-2a Example 3, Type 1 jitter, no noise shaping: Time domain distortion.

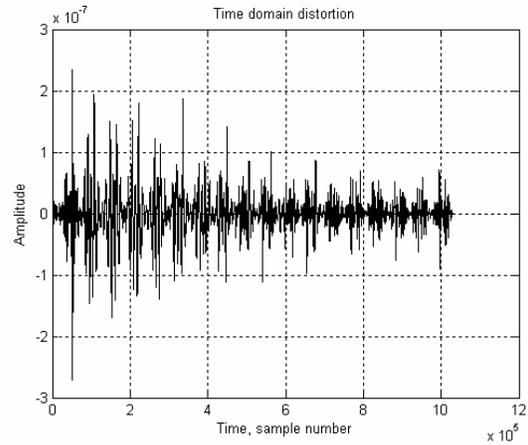


Figure 7-2d Example 3, Type 2 jitter, no noise shaping: Time domain distortion.

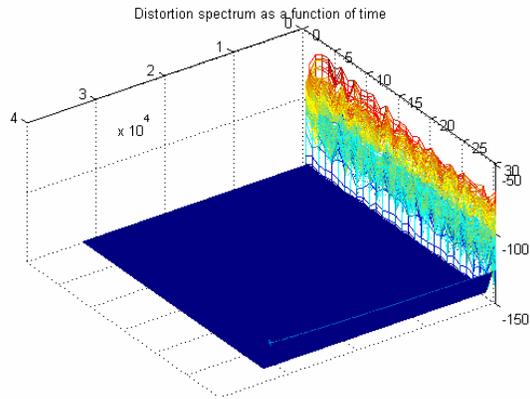


Figure 7-2b Example 3, Type 1 jitter, no noise shaping: Time-frequency spectrum.

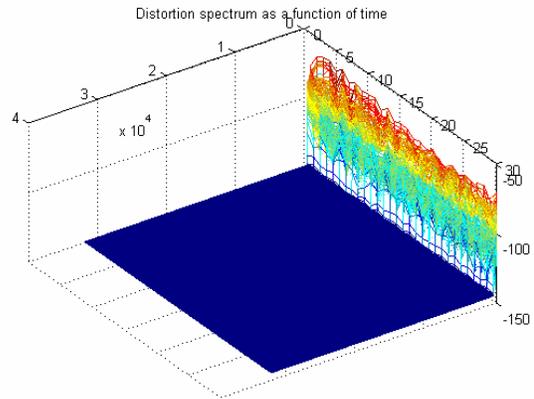


Figure 7-2e Example 3, Type 2 jitter, no noise shaping: Time-frequency spectrum.

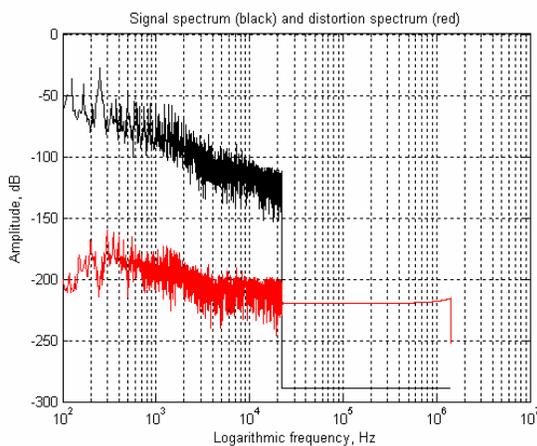


Figure 7-2c Example 3, order Type 1 jitter, no noise shaping: Signal and distortion spectra.

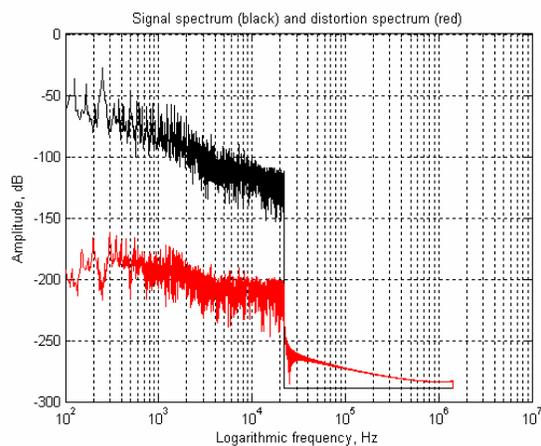


Figure 7-2f Example 3, Type 2 jitter, no noise shaping: Signal and distortion spectra.

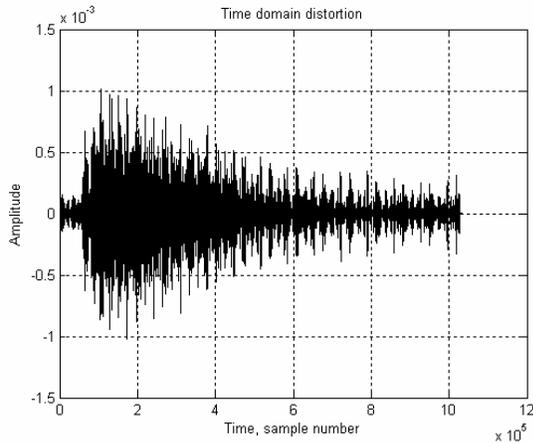


Figure 7-3a Example 1, Type 1 jitter, 2nd-order noise shaping 10 bit: Time domain distortion.

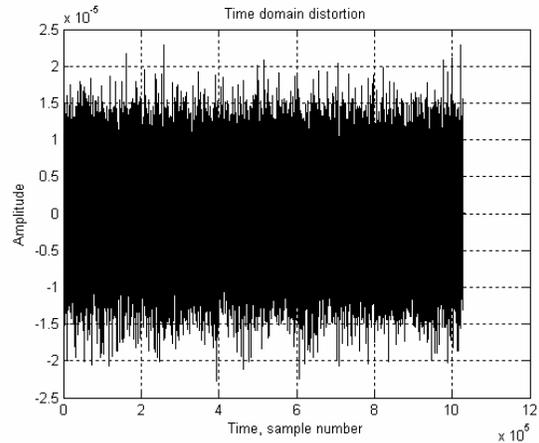


Figure 7-3d Example 1, Type 2 jitter, 2nd-order noise shaping 10 bit: Time domain distortion.

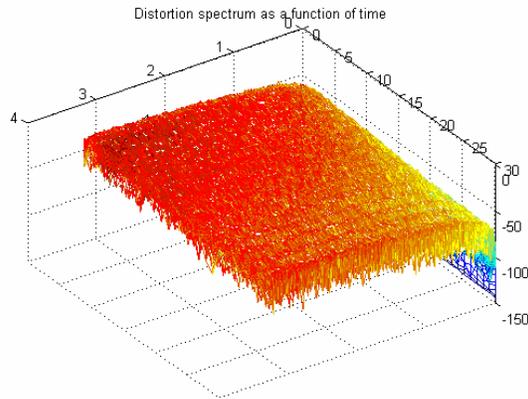


Figure 7-3b Example 1, Type 1 jitter, 2nd-order noise shaping 10 bit: Time-frequency spectrum.

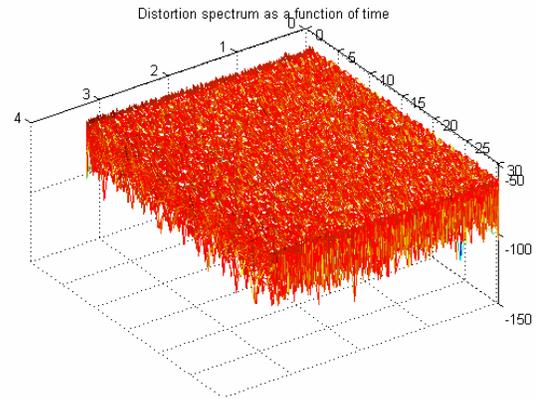


Figure 7-3e Example 1, Type 2 jitter, 2nd-order noise shaping 10 bit: Time-frequency spectrum.

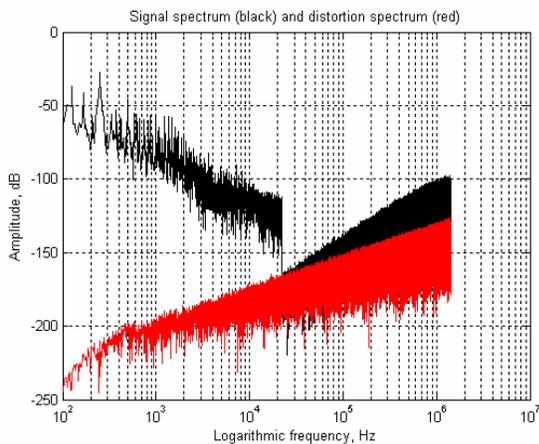


Figure 7-3c Example 1, Type 1 jitter, 2nd-order noise shaping 10 bit: Signal and distortion spectra.

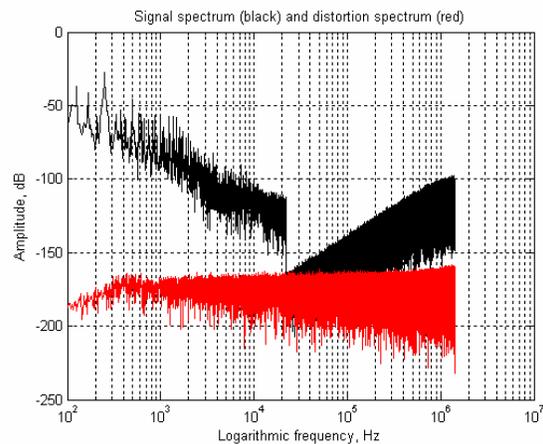


Figure 7-3f Example 1, Type 2 jitter, 2nd-order noise shaping 10 bit: Signal and distortion spectra.

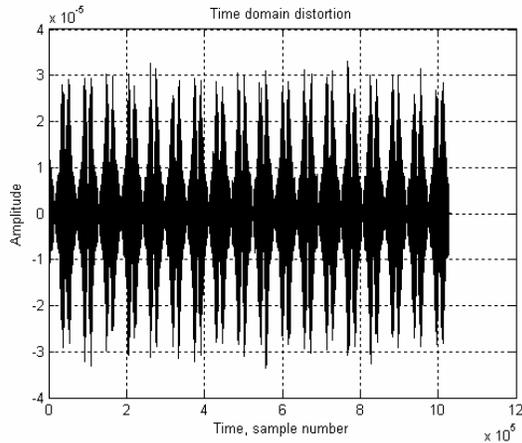


Figure 7-4a Example 3, Type 1 jitter, 2nd-order noise shaping 10 bit: Time domain distortion.

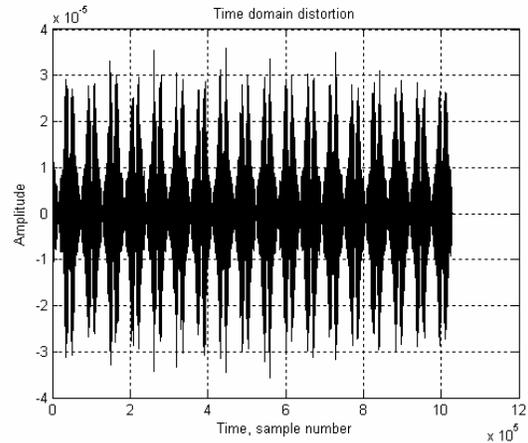


Figure 7-4d Example 3, Type 2 jitter, 2nd-order noise shaping 10 bit: Time domain distortion.

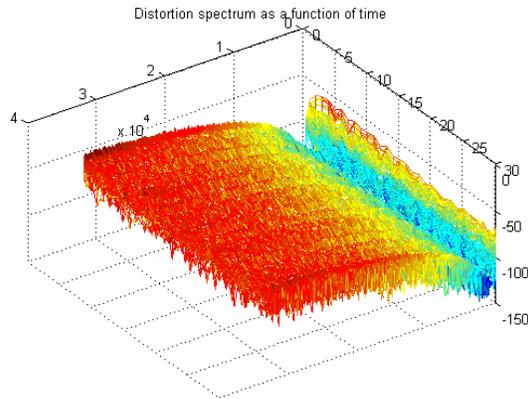


Figure 7-4b Example 3, Type 1 jitter, 2nd-order noise shaping 10 bit: Time-frequency spectrum.

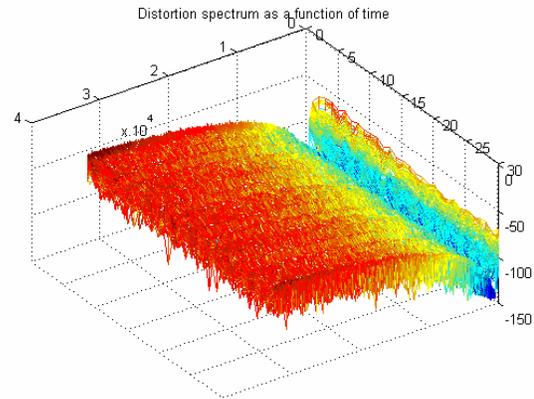


Figure 7-4e Example 3, Type 2 jitter, 2nd-order noise shaping 10 bit: Time-frequency spectrum.

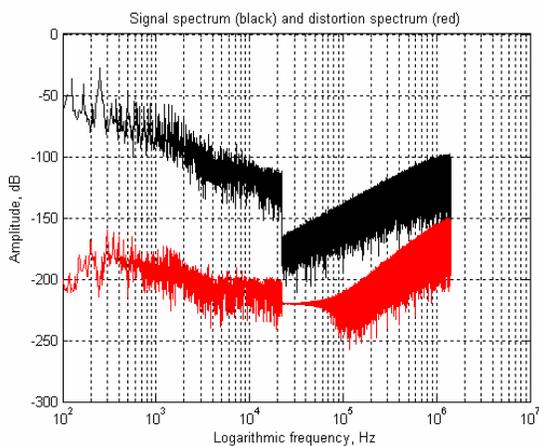


Figure 7-4c Example 3, Type 1 jitter, 2nd-order noise shaping 10 bit: Signal and distortion spectra.

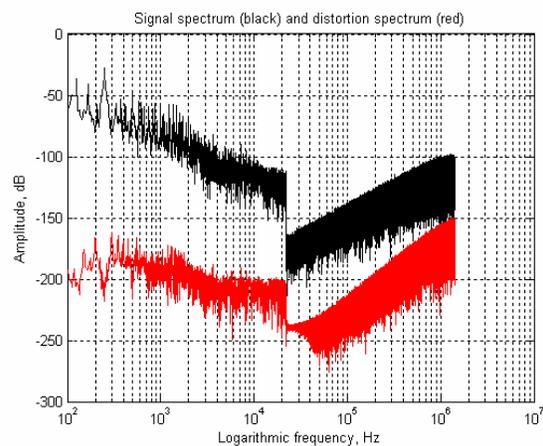


Figure 7-4f Example 3, Type 2 jitter, 2nd-order noise shaping 10 bit: Signal and distortion spectra.

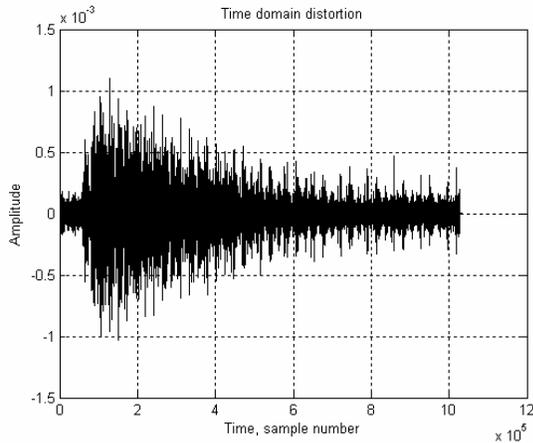


Figure 7-5a Example 1, Type 1 jitter, Sony-FF 6-bit noise shaping: Time domain distortion.

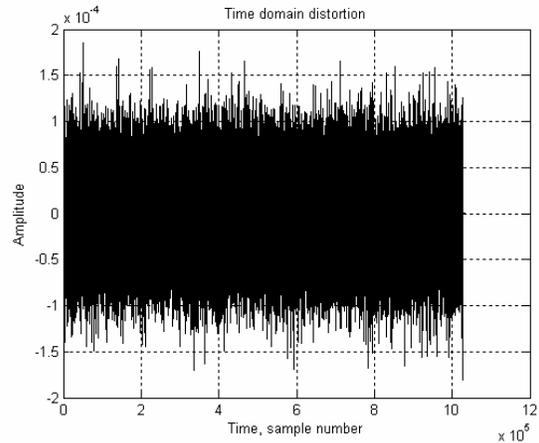


Figure 7-5d Example 1, Type 2 jitter, Sony-FF 6-bit noise shaping: Time domain distortion.

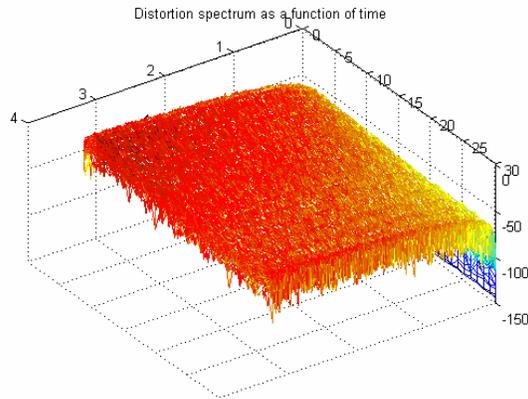


Figure 7-5b Example 1, Type 1 jitter, Sony-FF 6-bit noise shaping: Time-frequency spectrum.

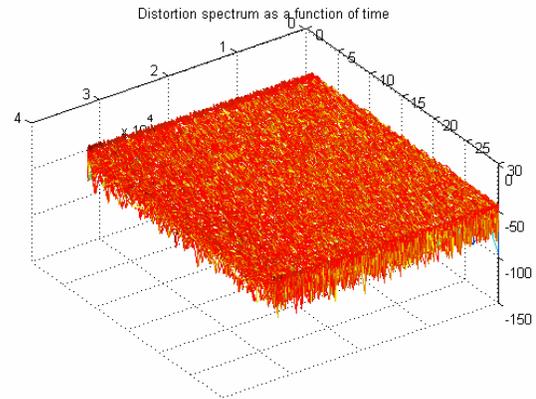


Figure 7-5e Example 1, Type 2 jitter, Sony-FF 6-bit noise shaping: Time-frequency spectrum.

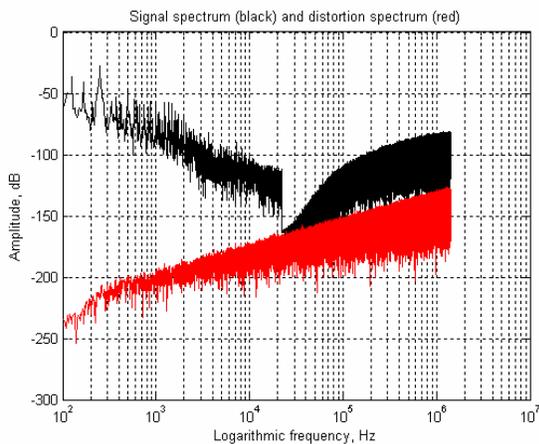


Figure 7-5c Example 1, Type 1 jitter, Sony-FF 6-bit noise shaping: Signal and distortion spectra.

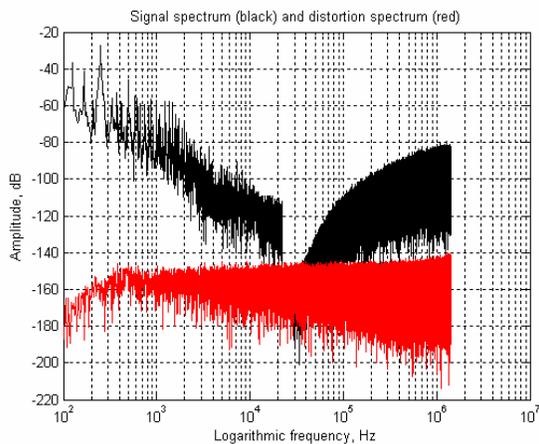


Figure 7-5f Example 1, Type 2 jitter, Sony-FF 6-bit noise shaping: Signal and distortion spectra.

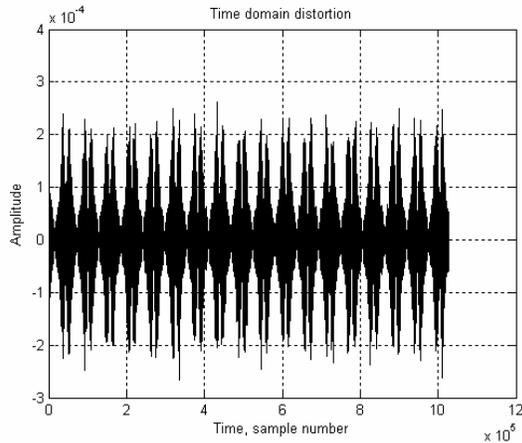


Figure 7-6a Example 3, Type 1 jitter, Sony-FF 6-bit noise shaping: Time domain distortion.

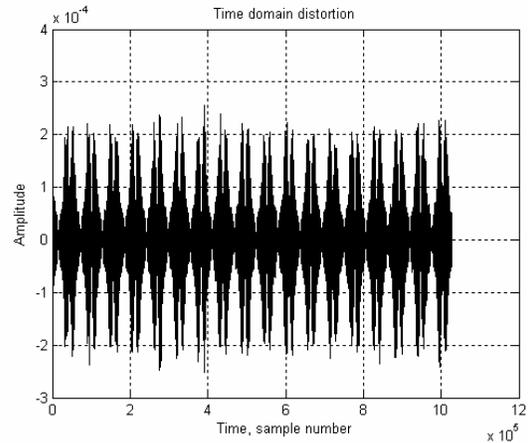


Figure 7-6d Example 3, Type 2 jitter, Sony-FF 6-bit noise shaping: Time domain distortion.

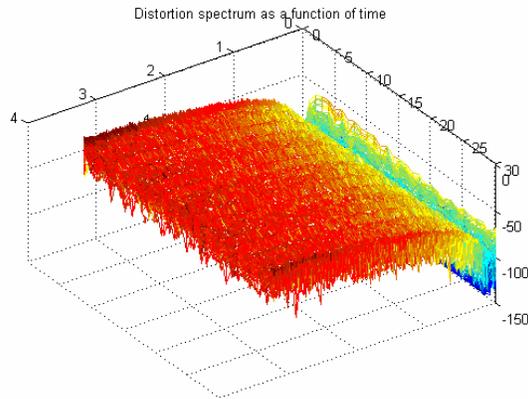


Figure 7-6b Example 3, Type 1 jitter, Sony-FF 6-bit noise shaping: Time-frequency spectrum.

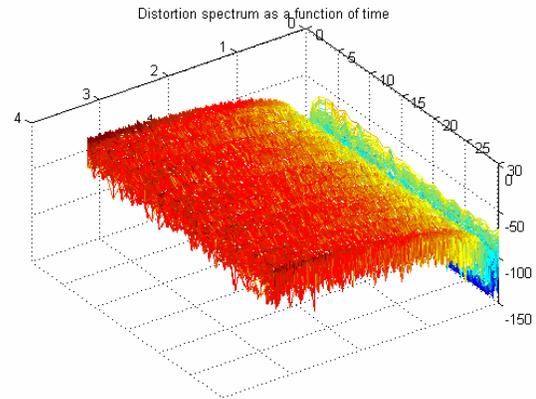


Figure 7-6e Example 3, Type 2 jitter, Sony-FF 6-bit noise shaping: Time-frequency spectrum.

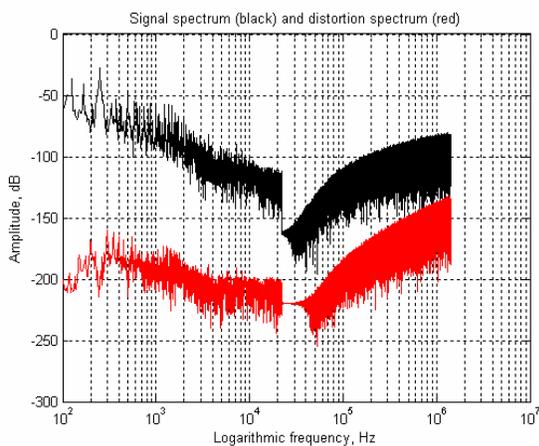


Figure 7-6c Example 3, Type 1 jitter, Sony-FF 6-bit noise shaping: Signal and distortion spectra.

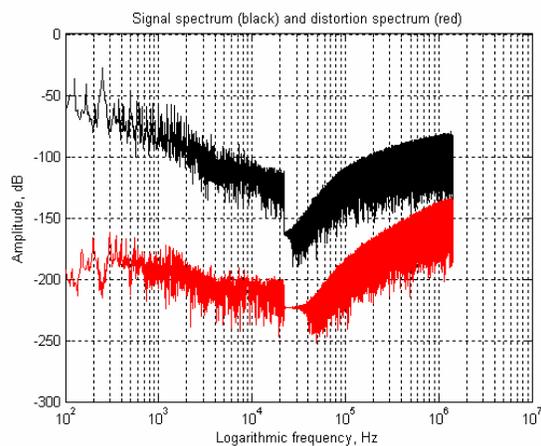


Figure 7-6f Example 3, Type 2 jitter, Sony-FF 6-bit noise shaping: Signal and distortion spectra.

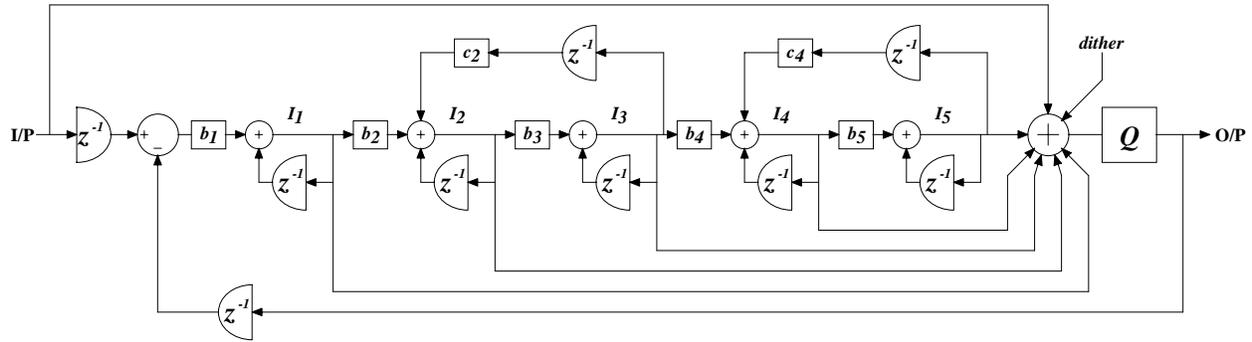


Figure 8-1 Sony-FF SDM discrete-time topology.

8. JITTER IN SDM SYSTEMS

The final system to be investigated in this study is SDM where for illustration the Sony-FF noise shaper [22] is employed using 2-level quantization to form a binary output sequence. The topology of this noise shaper is shown in Figure 8-1 where five cascaded integrators { $b_1=1$; $b_2=.5$; $b_3=.25$; $b_4=.125$; $b_5=.0625$ } are used enhanced by two regenerative local feedback loops { $c_2=-.001953125$; $c_4=-.03125$;} to improve high-frequency noise shaping by forming notches in the noise-shaping transfer function at 10 kHz and 20 kHz respectively. Although conceptually similar to the noise shaper used in Section 7 the restriction to 2-level quantization Q at the output implies that the output signal is always generated at a constant level even when the input signal is zero, which is not the case with multi-level quantization where the output only spans a few quanta under zero excitation. Consequently the system is always susceptible to the maximum impact of jitter with the expectation that random jitter will cause just a degradation in signal-to-noise ratio with minimal noise modulation artifacts. As a result it is normal to use output pulses of constant area by employing switched-capacitor techniques [17] where Type 1 jitter is the appropriate jitter description.

A similar test regime was employed to that presented in Section 7 where a 44.1 kHz @ 16 bit music signal was up-sampled by a factor of 64 to the Super-Audio CD (SACD) sample rate of 2.8224 Mbit/s [22,23]. The results are presented as follows:

Figure 8-2: Jitter → Example 1, Sony-FF noise shaping, output DAC resolution 1 bit

Figure 8-3: Jitter → Example 3, Sony-FF noise shaping, output DAC resolution 1 bit

In addition a further test was performed using the Miller sequence generated at 44.1 kHz @ 16 bit and up-sampled to 2.8224 Mbit/s prior to SDM encoding. The results of this test are presented as follows:

Figure 8-4: Jitter → Example 1, Sony-FF noise shaping, output DAC resolution 1 bit

Figure 8-5: Jitter → Example 3, Sony-FF noise shaping, output DAC resolution 1 bit

9. DISCUSSION

The results show the form of the distortion in the time domain for both random and periodic jitter. For the cases of random jitter then the distortion appears uniformly spread across the Nyquist bandwidth together with a weighting that depends upon whether the DAC is Type 1 or Type 2. Close inspection of the spectral plots reveal evidence of a low frequency roll-off, this process artifact is due to 64-times oversampling and the low-pass filter in the jitter simulator not being increased proportionally for computational reasons. However, for all cases that use a multi-level DAC then the jitter-induced noise is modulated in amplitude even though the long-term spectral average shows uniformity. Applying both oversampling and noise shaping spreads the jitter distortion over a wider bandwidth although the long-term spectral structure remains similar. As part of the experimentation regime, it was possible to audition just the distortion where it can be confirmed that modulation noise linked to the audio signal could be perceived although the effect of noise shaping and the resulting high frequency signals mitigated this to some extent. However, for the extreme case of 1-bit coding then the resulting distortion is mainly noise-like as amplitude modulation no longer occurs, this is an interesting consequence of using a 1-bit SDM code.

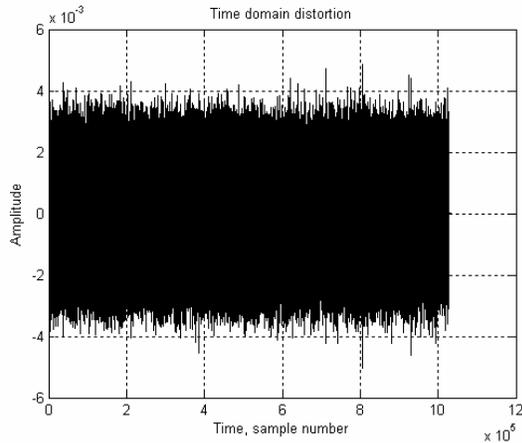


Figure 8-2a Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping: Time domain distortion.

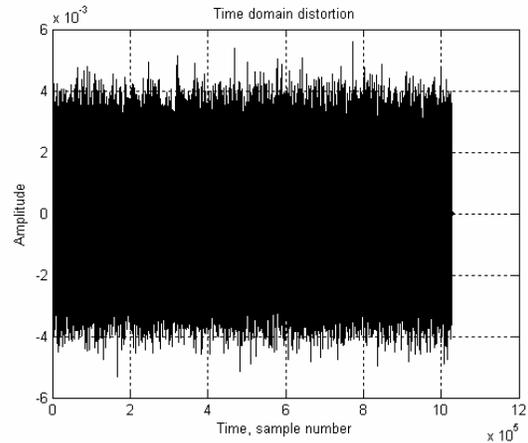


Figure 8-2d Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping: Time domain distortion.

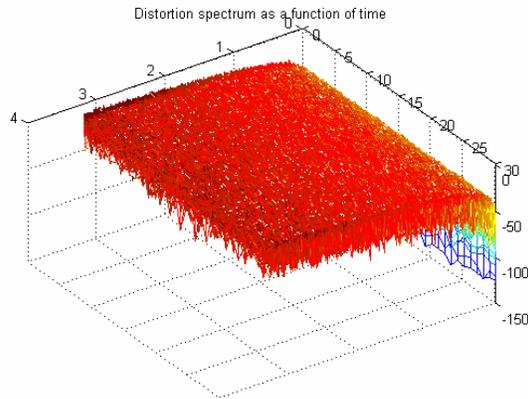


Figure 8-2b Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping: Time-frequency spectrum.

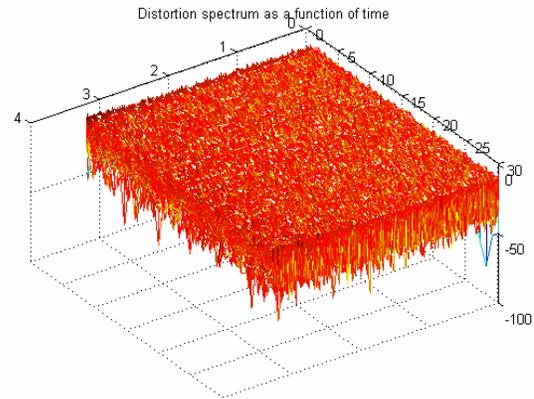


Figure 8-2e Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping: Time-frequency spectrum.

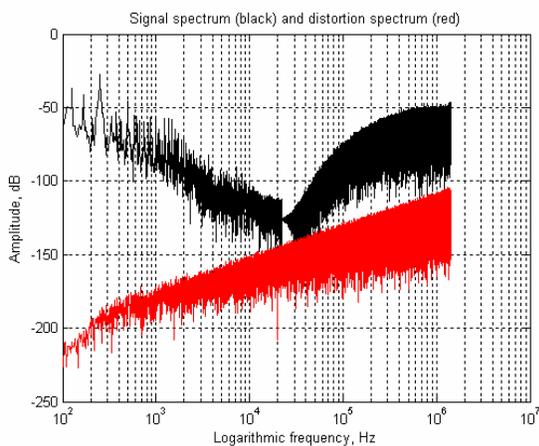


Figure 8-2c Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping: Signal and distortion spectra.

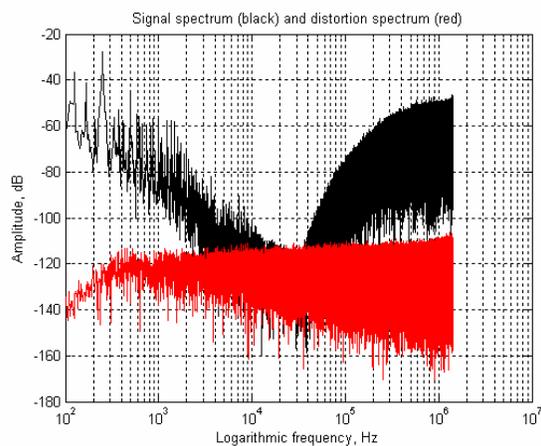


Figure 8-2f Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping: Signal and distortion spectra.

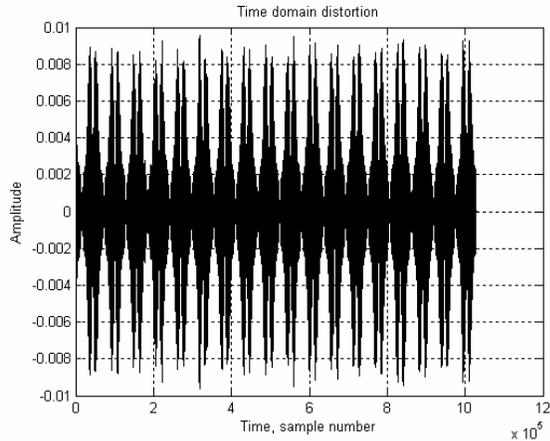


Figure 8-3a Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping: Time domain distortion.

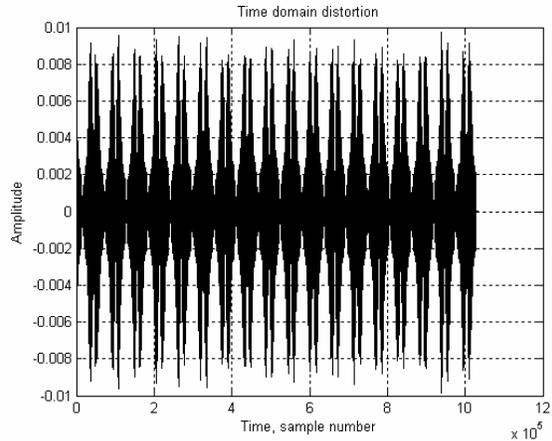


Figure 8-3d Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping: Time domain distortion.

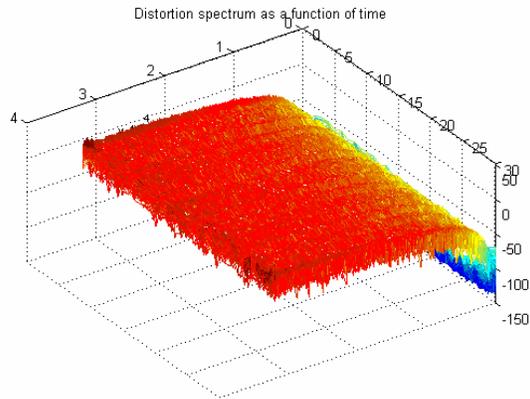


Figure 8-3b Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping: Time-frequency spectrum.

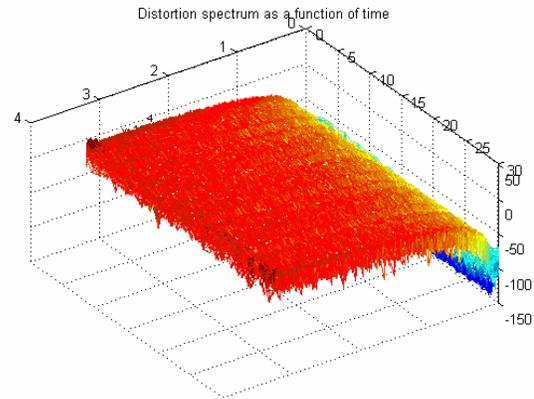


Figure 8-3e Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping: Time-frequency spectrum.

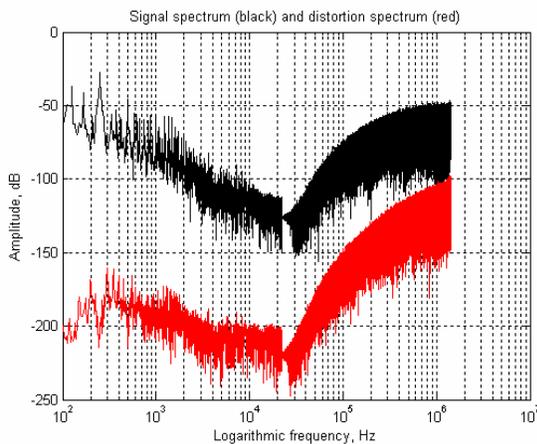


Figure 8-3c Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping: Signal and distortion spectra.

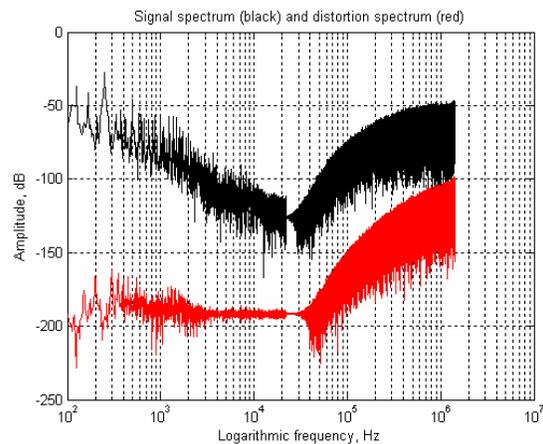


Figure 8-3f Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping: Signal and distortion spectra.

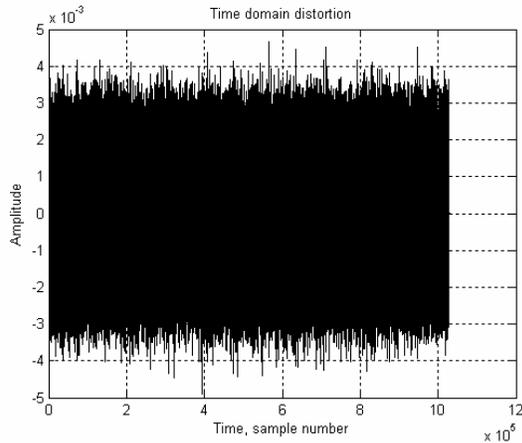


Figure 8-4a Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Time domain distortion.

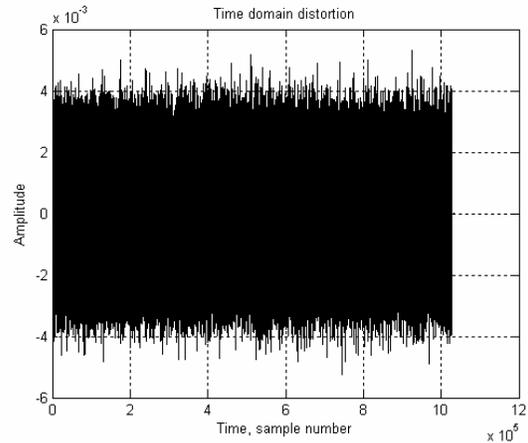


Figure 8-4d Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Time domain distortion.

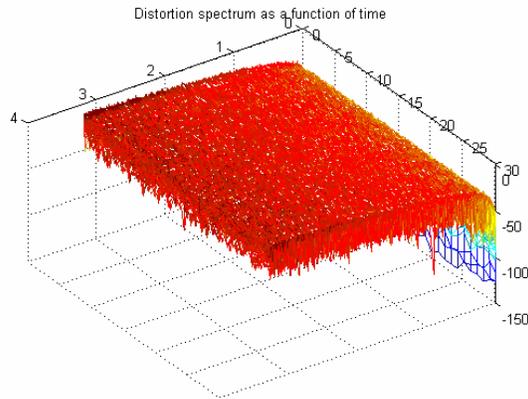


Figure 8-4b Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Time-frequency spectrum.

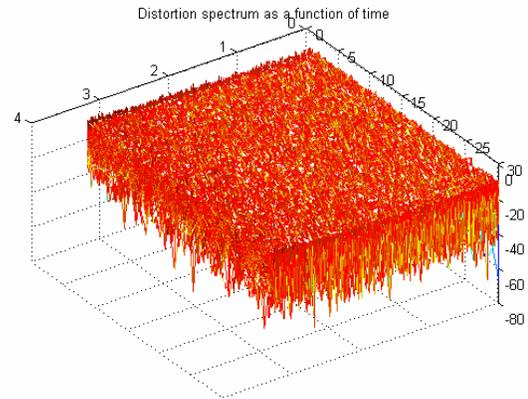


Figure 8-4e Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Time-frequency spectrum.

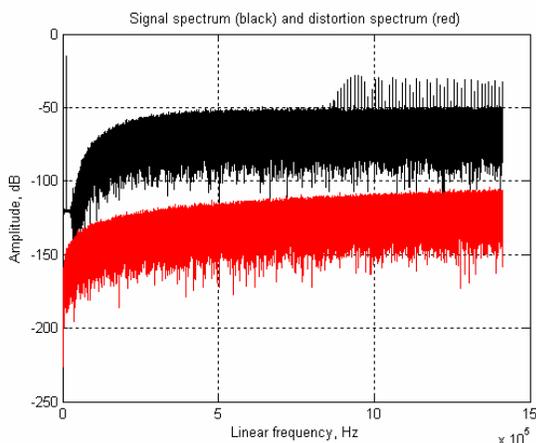


Figure 8-4c Example 1, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Signal and distortion spectra.

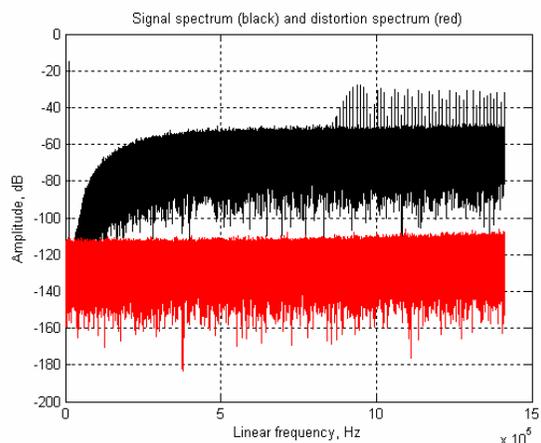


Figure 8-4f Example 1, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Signal and distortion spectra.

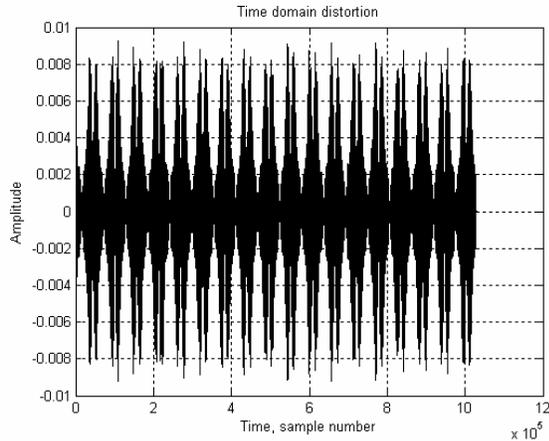


Figure 8-5a Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Time domain distortion.

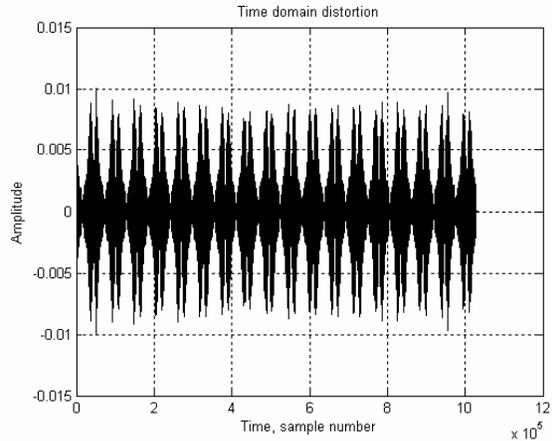


Figure 8-5d Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Time domain distortion.

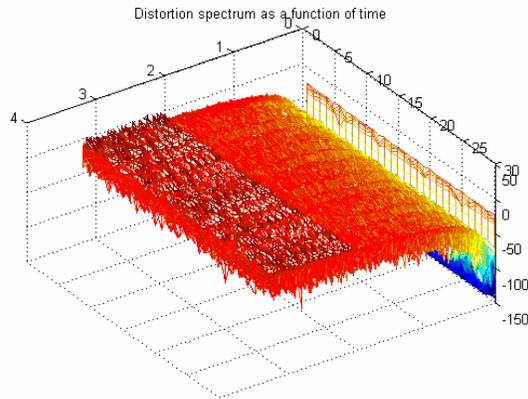


Figure 8-5b Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Time-frequency spectrum.

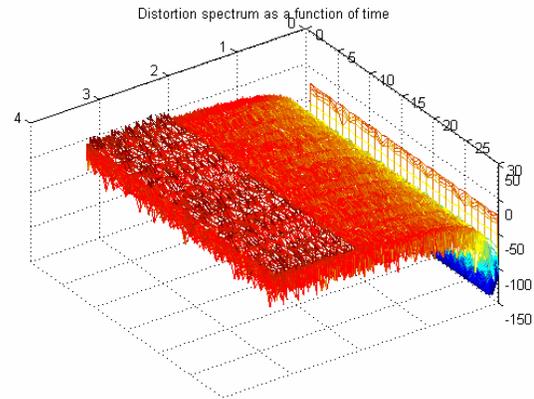


Figure 8-5e Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Time-frequency spectrum.

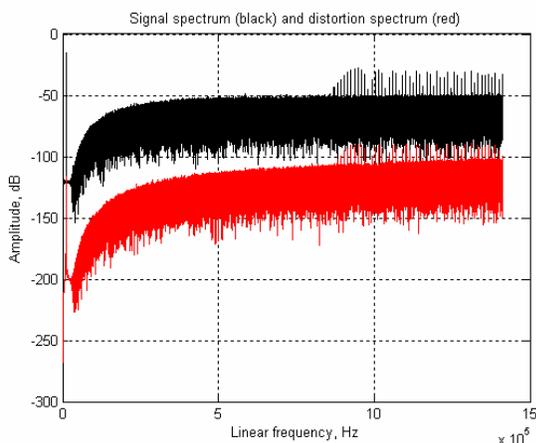


Figure 8-5c Example 3, Type 1 jitter, Sony-FF 1-bit noise shaping, Miller: Signal and distortion spectra.

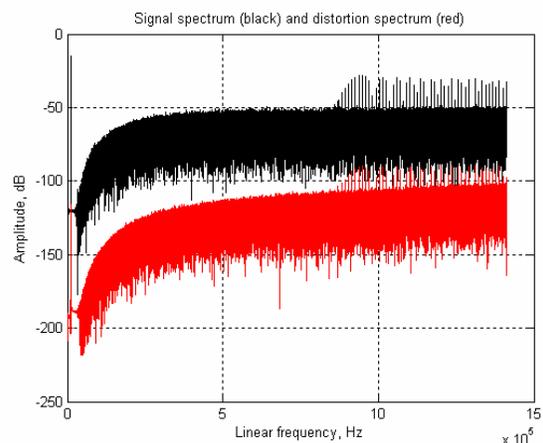


Figure 8-5f Example 3, Type 2 jitter, Sony-FF 1-bit noise shaping, Miller: Signal and distortion spectra.

However, changing the jitter to a periodic sequence (Example 3) revealed very different spectral domain results. For the cases of oversampling and noise shaping, the characteristic of the high frequency noise was retained as well as the generation of in-band intermodulation distortion. However, there is little evidence of the high frequency noise being reflected into the audio band.

10. CONCLUSIONS

Jitter is an important aspect of digital audio system design and as suggested by the simulations described, it can result in distortion that has a relatively complicated form. As stated, there are several mechanisms that give rise to jitter where in practice it is the relationship between jitter and signal that is critical. The paper has focused mainly on two types of jitter namely random and periodic although there are also mechanisms whereby jitter has correlation with the audio signal while in practice jitter can be a complicated mix of all these types. However, the type of converter also influences the way in-band distortion relates to the audio signal where both the presence of oversampling and noise shaping and whether a Type 1 or Type 2 output stage is used determines the form of the distortion.

The model presented here was designed to run at the selected sampling rate of the DAC which could include up-sampling and noise shaping and allow DAC amplitude resolutions to be accommodated that spans a range of 1 bit up to the quantization of the input file. As well as including modeling for both Type 1 and Type 2 DACs, a FIR filter was incorporated to prevent aliasing distortion from contaminating the distortion. Also, the method included an approximation to a stationary $\text{sinc}(x)$ function within the process of convolution which speeded computation and was possible because in general jitter is a small fraction of a sample period.

11. REFERENCES

1. The effects of sampling clock jitter on Nyquist sampling analog-to-digital converters, and on oversampling delta-sigma ADCs, Harris, S., *JAES*, vol. 38, no. 7/8, pp. 537-542, July 1990
2. Is the AES/EBU/SPDIF digital audio interface flawed?, Dunn, C. and Hawksford, M.O.J., *93rd AES Convention*, San Francisco, preprint 3360, October 1992
3. Jitter: Specification and assessment in digital audio equipment, Dunn, J., *93rd AES Convention*, October 1992, preprint 3361
4. Jitter analysis of asynchronous sample-rate conversion, Adams, R., *95th AES Convention*, October 1993, preprint 3712
5. The diagnosis and solution of jitter-related problems in digital audio systems, Dunn, J. and Dennis, I., *96th AES Convention*, February 1994, preprint 3868
6. Analysis of jitter rejection of SRCs and DACs using an NCO, Wood, M., *100th AES Convention*, May 1996, preprint 4261
7. Theoretical and audible effects of jitter on digital audio quality, Benjamin, E. and Gannon, B., *105th AES Convention*, September 1998, preprint 4826 (P-1)
8. Jither: The effects of jitter and dither for 1-bit audio PWM signals, Floros, A.C., Mourjopoulos, J. N. and Tsoukalas, D.E., *106th AES Convention*, May 1999, preprint 4956
9. Quantization and dither: A theoretical survey, Lipshitz, S.P. Wannamaker, R.A. and Vanderkooy, J., *JAES*, vol. 40, no. 5, 355-375; May 1992
10. Chaos, oversampling and noise shaping in digital-to-analog conversion, Hawksford, M.O.J., *JAES*, vol. 37, no. 12, pp 980-1001, December 1989
11. Digital-to-analog converter with low intersample transition distortion and low sensitivity to sample jitter and transresistance amplifier slew rate, Hawksford, M.O.J., *JAES*, vol. 42, no. 11, pp 901-917, November 1994
12. Current-steering transimpedance amplifiers for high-resolution digital-to-analogue converters, Hawksford, M.O.J., *109th AES Convention*, Los Angeles, September 2000, preprint 5192
13. A unity bit coding method by negative feedback, Inose, H. and Yasuda, Y., *Proc. IEEE*, vol. 51, p. 1524, November 1963
14. Analytical derivation of audio PWM signals and spectra, Floros, A.C. and Mourjopoulos, J.N.,

JAES, vol. 46, no. 7/8, pp 621-633, July/August 1998

15. Reeves, A.H., (1) French patent 852,183 (1930), also BP 535,860 (1939) and USP 2,272,070 1942, (2) The past present and future of PCM, *IEEE Spectrum*, 58-63, May 1965, (3) BP 1,046,792
16. Dynamic jitter filtration in high resolution DSM and PWM digital-to-analogue conversion, Hawksford M.O.J., *96th AES Convention*, Amsterdam, preprint 3811, February 1994
17. A CMOS stereo 16-bit converter for digital audio, D.J., Dijkmans, E.C., Stikvoort, E.F., McKnight, A.J., Holland, D.J. and Bradinal, W., *IEEE J., Solid-State Circuits*, SC-22, No. 3, June 1987
18. Clean clocks once and for all?, Frandsen, C.G. and Travis, C., *120th AES Convention*, Paris, 2006, paper 6700
19. <http://www.milleraudioresearch.com/index.html>
20. System measurement and identification using pseudorandom filtered noise and music sequences, Hawksford, M.O.J., *JAES*, vol. 53, no. 4, pp. 275-296, April 2005
21. A 116 dB SNR multi-bit noise shaping DAC with 192 kHz sampling rate, Adams, R., Nguyen, K. and Sweetland, K., *106th AES Convention*, Munich, 1999, preprint 4963 S5
22. Investigation of practical 1-bit delta-sigma conversion for professional audio applications, Takahashi H. and Nishio, A., *110th AES Convention*, Amsterdam, May 2001, paper 5392
23. Super Audio CD format, Verbakel, J., van de Kerkhof, L., Maeda, M. and Inazawa, Y., *104th AES Convention*, Amsterdam, May 1998, paper 4705

12. ACKNOWLEDGEMENT

I would like to offer my appreciation to the constructive comments and encouragement to publish this work made by Vicki Melchior while in the preparation of this paper.

13. APPENDIX: MATLAB JITTER SIMULATOR

The following Matlab code can be used to simulate jitter and calculate the resultant jitter distortion. It uses an external audio file named 'Source' and six examples of jitter are specified. Various option switches are included in the program introduction, which include selecting noise shaping options and output resolution.

```
% Type 1 and 2 jitter simulator
% jittertest.m
% noise shaping options
% external input file option
% place external audio wav file 'Source' in: cd C:\Temp
% 24.7.06

% initialise
home; clc; clear; colordef white; close
fprintf('Type 1 and 2 jitter simulator with noise shaping
options\n')

% select jitter model: Type 1 (type = 1) or Type 2 (type
= 2)
type=2;

% select jitter example, ex
ex=1;

% first and second-order + Sony ff sdm
% G=0; no noise shaping
% G=1: 1st-order loop multi-level
% G=2: 2nd-order loop multi-level
% G=3: Sony ff multi-level
% G=4: Sony ff SDM 1 bit
G=3;

% noise shaper output resolution (bits)
bits=6;

% input type: 0 external wav file, 1 internal sinusoidal
inp=1; % external file

% total vector length
L=2^20;

% over sampling ratio
over=64;

% peak jitter amplitude, second
jn=10^-9;
```

```

% number of coefficients in jitter re-sampling filter, odd
(2^12+1)
nc=2^13+1;

% read input wav offset
off=30000;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% jitter frequencies f1, f2, f3
% relative jitter amplitudes a1, a2, a3
% relative random jitter amplitude ar
if ex==1
f1=100; f2=901; f3=3003;
a1=0; a2=0; a3=0; ar=1;
elseif ex==2
f1=44100+50; f2=44100-50; f3=3003;
a1=1; a2=-1; a3=0; ar=0;
elseif ex==3
f1=50; f2=100; f3=150;
a1=1; a2=.5; a3=.25; ar=0;
elseif ex==4
f1=.2; f2=44100-50; f3=3003;
a1=0; a2=1; a3=0; ar=0;
elseif ex==5
f1=.2; f2=10; f3=3003;
a1=0; a2=1; a3=0; ar=0;
elseif ex==6
f1=.2; f2=44100-50; f3=3003;
a1=0; a2=0; a3=0; ar=0;
else; return; end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% read wav file
cd C:\Temp
len=round(L/over);
if inp==0
[y0,fs,bits]=wavread('Source', [off+1 off+len]);
else
ft=10000;
y0=sin(2*pi*(over*fs*round(L*ft/(over*fs)))/L)*(1:len)/(over*fs);
end

% set oversampling parameters (L total vector length,
len sub-sampled vector length)
fsam=over*fs;
f0=fsam/L;

% interpolate input to a sampling rate of fsam
fprintf('Interpolate input to fsam\n')

yi=zeros(1,L);
for x=1:len
yi((x-1)*over+1:x*over)=over*[y0(x) zeros(1,over-1)];
end
win=[ones(1,len/2) zeros(1,(L-len)/2)];
win=[win 0 win(L/2:-1:2)];
yi=real(ifft(fft(yi).*win));

% low-order noise shaping
rd=rand(1,L)+rand(1,L)-1;
y=zeros(1,L); I1=0; I2=0;
nl=2^(bits-1); % number of levels in noise shaper
output
if G==0 % no noise shaper
fprintf('No noise shaper\n')
y=yi;
elseif G==1 % first-order noise shaper
fprintf('First-order noise shaper\n')
for n=2:L
I1=I1+yi(n)-y(n-1);
y(n)=round(rd(n)+I1*nl)/nl;
end
elseif G==2 % second-order noise shaper
fprintf('Second-order noise shaper\n')
for n=2:L
er=yi(n)-y(n-1);
I1=I1+er;
I2=I1+I2+er;
y(n)=round(rd(n)+I2*nl)/nl;
end
elseif G==3 % Sony FF multi-level noise shaper
fprintf('Sony FF multi-level noise shaper\n')
rd=.35*rd;
b1=1; b2=.5; b3=.25; b4=.125; b5=.0625;
c2=-.001953125; c4=-.03125;
y=zeros(1,L); I=zeros(1,5);
for n=2:L
I(1)=I(1)+yi(n-1)-y(n-1);
I(2)=I(2)+b2*I(1)+c2*I(3);
I(3)=I(3)+b3*I(2);
I(4)=I(4)+b4*I(3)+c4*I(5);
I(5)=I(5)+b5*I(4);
y(n)=round((sum(I(1:5))+yi(n))*nl+rd(n))/nl;
end
elseif G==4 % Sony FF SDM 1-bit noise shaper
fprintf('Sony FF SDM 1-bit noise shaper\n')
rd=.35*rd;
b1=1; b2=.5; b3=.25; b4=.125; b5=.0625;
c2=-.001953125; c4=-.03125;
y=zeros(1,L); I=zeros(1,5);
for n=2:L
I(1)=I(1)+yi(n-1)-y(n-1);

```

```

I(2)=I(2)+b2*I(1)+c2*I(3);
I(3)=I(3)+b3*I(2);
I(4)=I(4)+b4*I(3)+c4*I(5);
I(5)=I(5)+b5*I(4);
y(n)=sign(sum(I(1:5))+yi(n)+rd(n));
end;
else
return; end

% compute jitter distortion
dist=zeros(1,L);
nc1=nc-1;
nc2=nc1/2;
dec=.5*(-1)^(nc2:-1:1).*ones(1,nc2)./(nc2:-1:1);
jit=[-dec 0 dec(nc2:-1:1)];
fprintf('Jitter generation\n')
rn=a1*sin(2*pi*f1*(1:L)/fsam)+a2*sin(2*pi*f2*(1:L)/fsam)+a3*sin(2*pi*f3*(1:L)/fsam)+ar*(rand(1,L)-rand(1,L));
rn=rn/max(abs(rn)); % normalize
rn=jn*rn*fsam; % scale
if type==1
fprintf('Non-linear convolution: Type 1\n')
for x=1:L-nc1
dist(x:x+nc1)=dist(x:x+nc1)+y(x)*rn(x)*jit;
end
else
fprintf('Non-linear convolution: Type 2\n')
for x=2:L-nc1
dist(x:x+nc1)=dist(x:x+nc1)+(y(x)-y(x-1))*rn(x)*jit;
end
for x=2:L
dist(x)=dist(x)+dist(x-1);
end; end

% derive time domain sequences of output signal and error (block length 25 ms)
fprintf('Calculate perceptual analysis window\n')
w=2^round(log10(.025*fsam)/log10(2));
winc(1:w)=0.5*(1-cos(2*pi*((1:w)-.5)/w));
mb=-1+2*L/w; % set number of analysis blocks
if mb>-1+2*L/w
mb=-1+2*L/w
end

% compute block transforms for distortion
fprintf('Computing block transforms\n')
b=1;
dist1=zeros(mb,w/2);
while b<mb
ss=abs(fft(dist(b*w/2:b*w/2+w-1).*winc(1:w)));
dist1(b,1:w/2)=20*log10(ss(1:w/2)+2^-24);

b=b+1;
end

% plot time domain audio waveform
plot(y(2*nc:L),'k')
title('Time domain source signal')
xlabel('Time, sample number')
ylabel('Amplitude')
grid, pause; close

% plot time domain audio waveform
plot(dist(2*nc:L),'k')
title('Time domain distortion')
xlabel('Time, sample number')
ylabel('Amplitude')
grid, pause; close

% plot 3-D spectral distortion
view(-35,130)
hold; grid
mesh(dist1(1:mb-1,1:w/2))
title('Distortion spectrum as a function of time')
pause; close

% plot signal and distortion spectra
fl=100; low=round(L*fl/fsam)+1;
fh=fsam/2; high=round(L*fh/fsam);
df=abs(fft(dist(1:L)))/L;
sf=abs(fft(y(1:L)))/L;
df=20*log10(df(1:L/2)+2^-48);
sf=20*log10(sf(1:L/2)+2^-48);
plot(fsam*(low:high)/L,sf(low:high),'k')
hold
plot(fsam*(low:high)/L,df(low:high),'r')
title('Signal spectrum (black) and distortion spectrum (red)')
xlabel('Linear frequency, Hz')
ylabel('Amplitude, dB')
grid, pause; close
semilogx(fsam*(low:high)/L,sf(low:high),'k')
hold
semilogx(fsam*(low:high)/L,df(low:high),'r')
title('Signal spectrum (black) and distortion spectrum (red)')
xlabel('Logarithmic frequency, Hz')
ylabel('Amplitude, dB')
grid, pause; close

% end program
return
%
*****

```