NONUNIFORM-TO-UNIFORM DECIMATION FOR DELTA-SIGMA FREQUENCY-TO-DIGITAL CONVERSION

W. Huff  I. Galton
UNIVERSITY OF CALIFORNIA, SAN DIEGO
9500 Gilman Dr, La Jolla, CA 92093-0407
whuff@ece.ucsd.edu  galton@ece.ucsd.edu

ABSTRACT

Recently, a delta-sigma frequency-to-digital converter (FDC) capable of 14-bit audio bandwidth data conversion at a 10MHz intermediate frequency (IF) was presented. The nonuniform-to-uniform decimation filter, a key component of this FDC, corrects for the nonuniform sampling inherent to the FDC and performs decimation filtering. This paper contains a detailed description of the nonuniform-to-uniform decimation filter and measured performance results for the entire FDC.

1. INTRODUCTION

DIGITAL signal processing is increasingly used in place of analog processing in wireless communication systems to reduce manufacturing costs, improve reliability, and allow programmability, particularly in high-volume, consumer-oriented systems. In receivers, the requisite A/D conversion is usually performed after the radio frequency (RF) signal has been down-converted to an IF in the zero to 100MHz range, and the demodulation is performed digitally. Although conventional ADCs are most commonly used for this purpose, the majority of wireless signal formats are based on frequency or phase modulation. In these cases, an alternative approach is to use a delta-sigma frequency-to-digital converter (ΔΣFDC)—a device that digitizes instantaneous frequency rather than amplitude—instead of an ADC.

Recently, a CMOS delta-sigma phase-locked loop (ΔΣPLL) has been demonstrated [1] that promises to make this an attractive approach. The ΔΣPLL is the key front-end component of a high-performance ΔΣFDC capable of 14-bit audio bandwidth data conversion at a 10MHz IF. To the knowledge of the authors, this level of performance has yet to be achieved using other FDC or bandpass ADC architectures in CMOS circuit technology [2], [3].

A high-level block diagram of the ΔΣPLL-based ΔΣFDC is shown in Figure 1. It consists of a hard limiter, a ΔΣPLL, a digital lowpass filter, a digital nonuniform-to-uniform downsampler, and a conventional digital decimation filter. The theory underlying the ΔΣPLL has been presented in [4], and the implementation details of the CMOS prototype mentioned above have been presented in [1]. The components following the ΔΣPLL in Figure 1 are collectively referred to as the nonuniform-to-uniform decimation filter. The details of the nonuniform-to-uniform decimation filter have not been presented previously and are the focus of this paper.

The ΔΣPLL performs an instantaneous frequency estimate each time that a positive going zero-crossing of the input signal occurs. Thus, the sampling rate of the ΔΣPLL is determined by the input frequency, so a frequency modulated (FM) input signal results in nonuniform sampling intervals.

Many techniques for reconstructing a signal from nonuniformly spaced samples have been described, such as generalized sampling theorems and Lagrange interpolation [5]. However, most of these methods are computationally expensive. Moreover, the quantization noise-shaping inherent to the ΔΣPLL is not generally present in other nonuniform-to-uniform conversion applications. The approach described herein achieves high-precision with very low implementation complexity by combining the nonuniform-to-uniform conversion operation with the decimation filtering required to remove the out-of-band quantization noise. The specific nonuniform-to-uniform decimation filter described is compatible with the ΔΣPLL prototype of [1] and gives rise to the above-mentioned ΔΣFDC performance, but the results are easily generalized to other ΔΣPLL configurations and ΔΣFDC specifications.

2. BACKGROUND

The ΔΣFDC input signal is assumed to have the form

\[ x(t) = A_c \sin(\omega_c t + \theta(t)) \]

where \( A_c \) is the amplitude, \( \omega_c \) is the carrier frequency, and \( \theta(t) \) is the instantaneous phase of \( x(t) \) relative to \( \omega_c t \). The instantaneous frequency of \( x(t) \) relative to \( \omega_c \) is \( \psi(t) = \frac{d\theta}{dt} \). For the system considered in the paper, it is assumed that \( \psi(t) \) is an audio bandwidth signal.

The ΔΣPLL converts the instantaneous frequency of its input signal into a digital output sequence with shaped quantization noise in a manner analogous to the way that a ΔΣ modulator converts the amplitude of its input signal into a digital output sequence. A functional diagram of the specific ΔΣPLL prototype presented in [1] is shown in Figure 2. It is clocked by an 80 MHz uniform master clock, \( \text{clk}_m \), and operates on a hard limited version of the input signal, \( x_n(t) \). The \( y[n] \) output signal is a quantized, nonuniformly sampled estimate of the instantaneous frequency of \( x(t) \) with quantization noise-shaping as will be described shortly. The \( \text{clk}_n \) output signal is a clock signal wherein each rising edge indicates that new valid data bits are available at the \( y[n] \) output.

The internal timing of the ΔΣPLL is controlled by two events: 1) The rising edge of \( x_n(t) \), and 2) the rising edge of the “carry” signal. The 4-bit counter is driven by \( \text{clk}_m \), and the “carry” signal goes high on its terminal count. In normal operation, the output of the XOR gate goes high when \( x_n(t) \) goes high. This switches the positive current source, \( I_p \), on until the “carry” signal goes high at which time it is switched off. The negative
current source, \( I_{\text{n}} \), is switched on for a fixed duration each time the “carry” signal goes high. Two \( \text{clk}_n \) cycles after the “carry” signal goes high, filtered ADC data is loaded into the counter, effectively setting the time until the next carry. The “carry” signal and its time-shifted derivatives are synchronized to the actual sample times of the \( \Delta \Sigma \) PLL, and the “carry+1” signal is brought out from the \( \Delta \Sigma \) PLL as \( \text{clk}_n \). From the results in [4] it can be shown that the \( \Delta \Sigma \) PLL of Figure 2, with an appropriate LSB normalization, produces the output sequence

\[
y[n] = \frac{4}{\pi} \tilde{\psi}[n] + e_2[n],
\]

where \( e_2[n] \) is 2nd-order shaped quantization noise with power spectral density (PSD) proportional to \( \sin^4(\omega/2) \), and \( \tilde{\psi}[n] \) is an estimate of instantaneous frequency corresponding to the change in instantaneous phase between adjacent samples. That is, \( \tilde{\psi}[n] = \int_{n-1}^{n} \psi(\tau) d\tau \), where \( t_n \) is the time of the \( n \)th sample of \( \psi(t) \).

It can also be shown from the theory in [4] that \( y[n] \) can be written in terms of the sample times, \( t_n \), as

\[
y[n] = 8 \left(1 - \frac{\delta_n}{T_c}\right) + e_2[n],
\]

where \( T_c = 2\pi/\omega_c \) is the nominal period of the input signal, and \( \delta_n = t_n - t_{n-1} \).

The average sample-rate of the \( \Delta \Sigma \) PLL is equal to the center frequency of \( \psi(t) \) which can range between 8.9MHz and 10MHz, but the sampled signal, \( \psi(t) \), is an audio-bandwidth signal. Therefore, \( \tilde{\psi}[n] \) is oversampled by a factor of approximately 200, and resides at very low frequencies in the discrete-time spectrum. In contrast, due to its \( \sin^4(\omega/2) \) PSD shape, most of the quantization noise power resides at high frequencies.

The \( \Delta \Sigma \) PLL is similar to a \( \Delta \Sigma \) modulator in that both quantize an oversampled version of the input signal such that the quantization noise is spectrally shaped to reside at high frequencies. However, an important difference is that the nonuniform sampling of the \( \Delta \Sigma \) PLL causes a slight spectral spreading of the oversampled signal. As discussed below, this spreading effect places a design constraint on the passband flatness of the subsequent filter stage. The nonuniform sampling and resulting spreading effect are corrected for by the nonuniform-to-uniform decimation filter.

3. DECIMATION ARCHITECTURE

The nonuniform-to-uniform decimation technique takes advantage of the property that the output sequence from the \( \Delta \Sigma \) PLL can be used to estimate both \( \delta_n \) and \( \tilde{\psi}[n] \). From the sequence of \( \delta_n \) values, the actual sample times can be estimated, and from the sequence \( \tilde{\psi}[n] \), frequency estimates for desired sample times can be calculated by 2-point interpolation among actual sample times.

3.1 Comb Filtering

The \( \Delta \Sigma \) PLL output sequence, \( y[n] \), is filtered by \( H(z) \). For this implementation, \( H(z) \) is a \( \text{comb}^2 \) filter with a transfer function \( H(z) = \left( \frac{z - 1}{1 - z^{-1}} \right)^2 \). Its purpose is to remove as much of the out-of-band quantization noise as practical without significantly distorting the components of the oversampled signal that underwent spectral spreading as a result of the nonuniform sampling. Thus, passband flatness and stopband attenuation are both design constraints. In addition, since the discrete-time frequency of the desired stopband shifts with instantaneous carrier frequency, a wide stopband is desired. The particular choice of \( H(z) \) represents a good compromise with respect to these tradeoffs. Moreover, it can be implemented efficiently using a multiplier-free recursive structure.

The filtered sequence, \( y_f[n] \), still contains the low-frequency signal information, but with much of the high-frequency quantization noise removed [6]. Thus, from (1), (2), and the frequency response implied by \( H(z) \),

\[
y_f[n] = \frac{64}{\pi} \tilde{\psi}[n] + e_f[n],
\]

and

\[
y_f[n] = 128 \left(1 - \frac{\delta_n}{T_c}\right) + e_f[n],
\]

where \( e_f[n] \) is the residual quantization noise remaining after \( e_2[n] \) is filtered by \( H(z) \). The passband droop associated with \( H(z) \) has been neglected.

3.2 Nonuniform-to-Uniform Downsampler

The two sub-blocks comprising the nonuniform-to-uniform downsampler of Figure 1 are shown in detail in Figure 3. The uniform clock generator extracts the sample time information from \( y_f[n] \) to determine which pairs of actual sample times straddle the desired sample times. The interpolating downsampler performs frequency estimation at the desired sample times by two-point interpolation about these actual sample times. The desired sample times are defined to occur at uniform intervals of \( T_s \), where \( T_c = PT_s \), and \( P \) is the average downsampled ratio. For the implementation described in this paper, \( P = 4 \).

From Figure 3 and equation (4), it follows that

\[
r[u] = -128 \left( \frac{\delta_n}{T_c} \right) + 256 + e_f[n].
\]

The purpose of the register \( r[n] \) shown in Figure 3 is to accumulate time interval estimates. During normal operation, \( r[n] \) stores the value of the actual time estimate relative to the desired sample time. This value can be either positive or negative. If \( r[n] \) is initially negative, it can be seen from Figure 3 that the scaled output of the hard limiter (\( \text{clk}_n \)) is \(-256\), which cancels the +256 DC offset of \( r[n] \) from (5), and causes a value proportional to \( \delta_n \) to be added to \( t[n-1] \) thereby obtaining \( t[n] \). Since \( \delta_n \) is always positive by definition, \( t[n] \) will monotonically increase while \( \text{clk}_n \) is -1. After \( t[n] \) exceeds zero, the output of the hard limiter switches from -1 to 1, so the scaled hard limiter output will combine with the +256 DC offset of \( r[n] \), ensuring that the next
value of \( t[n] \) is negative. This effectively sets a new desired sample time. Thus, each rising edge of \( c[k] \) occurs when the accumulated time estimate first exceeds each desired sample time. At this time, from (5) and Figure 3, the relation between the actual time estimate and the desired sample time is

\[
t[n] = \frac{128}{T_c} (t_n - T_k) + e_t[n],
\]

where \( t_n \) is the actual sample time, \( T_k \) is the desired sample time, and \( e_t[n] \) is the quantization noise remaining at \( t[n] \).

The interpolating downsample sampler processes the information contained in \( y[n] \) to extract uniformly spaced estimates of the instantaneous frequency of \( x(t) \). The first processing stage in the interpolating downsample sampler accumulates \( y_j[k] \) to obtain \( \hat{\theta}[n] \). The \( n^{th} \) sample of the resulting sequence is the nonuniform phase estimate at time \( t_n \). Interpreting the LSB value to be \( \pi/64 \), the sequence of nonuniform phase estimates can be written as

\[
\hat{\theta}[n] = \pi/64 \sum_{k=1}^{n} y_j[k] + \hat{\theta}[0],
\]

where \( \hat{\theta}[0] \) is an arbitrary initial value.

As described above, the rising edge of the uniform clock \( c[k] \) occurs when the actual sample time first exceeds the next desired sample time, \( T_k \). The actual sample times \( t_{m+1} \) and \( t_m \) thus straddle the desired sample time. From the nonuniform phase estimates at \( t_{m+1} \) and \( t_m \), two-point interpolation is used to find the uniform phase estimate, \( \hat{\theta}_u[k] \), at time \( T_k \). This is similar to the two-point interpolation method introduced in [7], in which it was shown that the interpolation error is small provided the modulation frequency is much smaller than the carrier frequency, as is the case here.

The two-point interpolation expression is

\[
\hat{\theta}_u[k] = \hat{\theta}[n] - \left( \hat{\theta}[n] - \hat{\theta}[n-1] \right) \left( t_n - T_k \right) \left( t_n - t_{n-1} \right),
\]

which is depicted graphically in Figure 4. An expression equivalent to (8) can be derived from Figure 3, the definition for \( \delta_u \), and equations (4), (6), and (7). Neglecting the LSB normalization in (7), and neglecting quantization noise,

\[
\hat{\theta}_u[k] = \hat{\theta}[n] - \frac{t[n]}{128} \left( \frac{y_j[n]}{1 - \frac{128 y_j[n]}{128 y_j[n]}} \right).
\]

The system of Figure 3 implements this form of the two-point interpolation expression.

As seen in Figure 3, the \( 1 - z^{-1} \) operation performed upon \( \hat{\theta}_u[k] \) produces uniform frequency estimates, \( \hat{\psi}_u[k] \). Ignoring quantization noise and interpolation error, it follows that \( \hat{\psi}_u[k] = \delta_u[k] - \hat{\theta}_u[k-1] \), which is proportional to the phase difference between times \( T_{k-1} \) and \( T_k \). Thus, the sequence \( \hat{\psi}_u[k] \) corresponds to frequency estimates at uniformly spaced sampling intervals.

3.3 Conventional Decimation Filter

The final processing block shown in Figure 1 is a conventional 50-fold lowpass decimation and equalization filter. It removes the remaining out-of-band quantization noise and reduces the output sequence to the final desired sampling rate. Equalization filtering is incorporated in the block to compensate for the passband droop of \( H(z) \) as well as that imposed by the effective moving average of the desired signal performed by the nonuniform-to-uniform conversion processing.

4. EXPERIMENTAL RESULTS

The performance of the Delta-Sigma ADC with nonuniform-to-uniform decimation was measured by combining the existing Delta-Sigma prototype [1] with a software implementation of the nonuniform-to-uniform decimation filter. The bus widths and other processing details were implemented exactly as shown in Figures 1 and 3, and identical results would be expected from a fully integrated VLSI implementation.

The signal to noise and distortion ratio (SINAD) and spurious free dynamic range (SFDR) measurements were made using an input signal with a sinusoidal instantaneous frequency function: \( \psi(t) = 2 \pi f_s \Delta F \sin(2 \pi f_s t) \). Thus \( x(t) \) was a sinusoidally modulated FM signal with a peak frequency deviation of \( \Delta F \) and a modulation frequency of \( f_s \). Measurements were taken with numerous values of \( \Delta F \) and \( f_s \). The worst case results for peak SINAD and peak SFDR were 85dB and 88dB, respectively, indicating 14dB worst-case performance.

Representative examples of the measured data are shown in Figures 5 and 6. Figure 5 shows the PSD of \( \hat{\psi}_u[m] \) corresponding to \( \Delta F = 1 \) MHz and \( f_s = 8 \) kHz. The measured (non-peak) SINAD and SFDR for this example are 85dB and 88dB, respectively. Figure 6 shows measured values of SINAD and SFDR as a function of \( \Delta F \) (recall that \( \Delta F \) is to the Delta-Sigma what amplitude is to a Delta-Sigma) again for \( f_s = 8 \) kHz. In this case, the measured peak SINAD and peak SFDR are 89dB and 94dB.

An intermodulation measurement was made using an input signal with a two-tone instantaneous frequency function of the form: \( \psi(t) = 2 \pi F_1 \sin(2 \pi f_1 t) + 2 \pi F_2 \sin(2 \pi f_2 t) \). Figure 7 shows data corresponding to the PSD of \( \hat{\psi}_u[m] \) with \( \Delta F_1 = \Delta F_2 = 250 \) kHz, \( f_1 = 22.1 \) kHz, and \( f_2 = 24.3 \) kHz. As shown in the figure, the maximum in-band intermodulation (IM) product for this test was –87dB.
REFERENCES


Figure 1: ΔΣ frequency-to-digital converter (ΔΣFDC)

Figure 2: ΔΣPLL prototype functional diagram.

Figure 3: Nonuniform-to-uniform downsampler.

Figure 4: Interpolation between actual sample points to find desired sample points

Figure 5: Measured PSD for single tone test

Figure 6: Measured SINAD and SFDR vs. peak frequency deviation

Figure 7: Measured PSD for two tone test