Multiplierless FIR Filters With Discrete-Value Sigma-Delta Encoded Coefficients

Scott R. Powell, Paul M. Chau

University of California
Dept. of ECE, R-007
LaJolla, CA 92093-0407

Abstract

This paper presents an oversampled sigma-delta modulation based encoding scheme to reduce an FIR filter's coefficient or input signal samples to values with only a few non-zero bits eliminating the need for multiplications. This new scheme realizes a substantial reduction in sample rate and additions over previous differential digital coding schemes by replacing the direct truncation within a sigma-delta modulator with a more sophisticated quantizing procedure which truncates to the next nearest canonic signed-digit code (number representable as sum or difference of a few powers-of-two). Computer simulations are used to demonstrate and verify the expected performance of this scheme and comparisons are made with previously reported results.

1 Introduction

Over the past two decades, many filter structures have been proposed to reduce the hardware complexity of fixed-point digital FIR filters. Most have concentrated on reducing the number and/or wordlength of coefficient multipliers. Delta modulation [1], sigma-delta modulation [2], and error spectrum shaping [3] techniques use differential encoding schemes to move quantization errors out of the frequency band of interest so that either the coefficient or the input signal wordlength can be truncated to a small number of bits. These filters require no general purpose multipliers.

Another method is to use the full coefficient wordlength but only permit a few of the bits to be non-zero. The resulting sum of signed power-of-two space is also referred to as canonic signed-digit (CSD) code [6]. With this scheme, multiplications can be implemented with a few additions.

This paper combines the above two techniques so that the error introduced by rounding coefficients (or the input signal) to the next nearest CSD code is moved out of the frequency band of interest. Thus, unlike previous differential encoding schemes, the full coefficient wordlength is maintained but restricted to discrete CSD values. Also, unlike the straight CSD truncation schemes, the quantization error is moved out of the frequency band of interest.

2 CSD Sigma-Delta Truncation

Figure 1a repeats the discrete time block diagram of [2] showing a single-loop sigma-delta modulator with a uniform quantizer truncating a 10 bit input sequence into a 2 bit output sequence. Multiplication using this output can be implemented with a single addition. The output sequence can take on only four discrete values. Figure 1b shows the proposed sigma-delta encoding scheme with a non-uniform quantizer truncating a 10 bit input sequence into a 2 bit CSD code. Multiplication using this output requires only one addition, the same as the previous approach of figure 1a.

Figure 1: Single-Loop Sigma-Delta Modulators Used to Truncate 10-bit Inputs to 2-bit Outputs.
(a) Standard Approach - Uniform Quantizer
(b) Proposed Approach - CSD Quantizer.
Implementing a multiplication operation with samples encoded using uniform quantizer based sigma-delta truncation requires the same number of additions as with CSD based sigma-delta truncation. However, the 2 bit CSD output of figure 1b can take on 148 discrete values, 37 times more than the uniform quantizer case of figure 1a. In general, CSD based sigma-delta truncation has many times more possible discrete-values thereby offering substantially reduced quantization error over previous approaches with no increase in hardware for downstream multiplication operations.

A sinusoid (10 bit) has been applied to the previous 2-bit sigma-delta truncator of figure 1a and the proposed 2-bit CSD sigma-delta truncator of figure 1b. Figure 2 compares the spectral response at the output of each truncator. As expected, the increased number of discrete-values for the CSD case has reduced the overall quantization noise and increased the equivalent number of bits within a given bandwidth. The equivalent number of bits available for a given oversampling ratio (OSR) can be computed by integrating the power spectra of figure 2 to determine rms noise power. Figure 3 compares the equivalent number of bits available as a function of OSR and the number of bits of truncation bits (either CSD or uniform) for the uniform quantization and new CSD quantization sigma-delta encoding methods. As can be seen from figure 3, the CSD based sigma-delta encoding method can be used at a much lower OSR than the uniform quantization method for any given resolution. Thus, when the truncated output is used as FIR filter coefficients, the CSD based sigma-delta scheme requires many times fewer additions per filter output sample than previous approaches.

As described in references [1]-[3], differential encoding can be applied either to the input signal or to the filter's impulse response sequence. Although rounding to a CSD value is more difficult than rounding to a binary PCM value, sigma-delta modulation applied to a filter's impulse response is performed off-line where the increased rounding complexity is inconsequential. For applications where sigma-delta modulation is used to encode the input signal, circuits have been recently reported [5] which can perform the CSD rounding in real-time. Since only one sigma-delta modulator is required for this case, the percentage increase in hardware for CSD truncation over using straight truncation is relatively small.

3 FIR Filters Based on Sigma-Delta Encoding

A multiplier free FIR filter can be realized by oversampling (interpolating) the desired impulse response by OSR times, passing this sequence through the sigma-delta CSD truncator of figure 1b, and using the truncated output as the FIR filter coefficients. Note that this is a multi-rate approach and requires interpolation of the filter's input signal by OSR times as well as decimation of the filter's output signal by OSR times. The interpolation and decimation operations and implementations are described in detail in [2] and are not considered in this paper.

Assuming the CSD truncation within the sigma-delta loop (see figure 1b) is done offline using a look-up table, the cpu time required to perform coefficient truncation increases only linearly with filter length. Although the coefficient truncation is suboptimal, filters with virtually unlimited length can be
designed in seconds or minutes. Additionally, the truncation need not be limited to the sum of only 2 powers-of-two. Samueli [6] has shown that permitting some coefficients to have three or more powers-of-two can result in very efficient filters.

To use the CSD sigma-delta truncator of figure 1b to truncate the coefficients of an FIR filter, several implementation details must be addressed. First is input scaling. Since the normalized frequency characteristics of a filter are not affected by multiplying all coefficients by a constant value, we consider the input scaling factor to be a free parameter. For simplicity, we do not attempt to optimize the input scale factor in this paper and simply set it such that the largest filter coefficient value uses the full dynamic range available from the sigma-delta loop. For example, if the sigma-delta CSD truncator of figure 1b were to truncate to the nearest 16 bit binary value which had at most three non-zero bits (a 3 bit CSD code), we would choose to scale the filter coefficients by (assume two's complement):  

\[ S = \frac{2^{B-1} - 1}{h_{\text{max}}} \]  

Where:  
\[ S = \text{Input Scale Factor} \]  
\[ B = \text{Original Wordlength (16 bits)} \]  
\[ h_{\text{max}} = \text{Largest Coefficient Value} \]

![Original FIR Impulse Response → Periodic Extension](image)

Figure 4: Periodically Extended FIR Impulse Response

Since a sigma-delta loop contains feedback, the output will exhibit a starting and ending transient response when presented with a finite length impulse response as an input signal. To overcome this difficulty, we propose using an input consisting of a periodic extension of the FIR impulse response, as illustrated in figure 4. Periodic inputs to the sigma-delta loop result in periodic outputs. Assuming enough periods are applied to the input (usually one is sufficient), the truncated FIR coefficient values are determined by extracting an appropriate portion of the sigma-delta loop output after steady-state is reached.

Although the output of the sigma-delta CSD truncation loop will be periodic, it will not necessarily be symmetric, even though the input is symmetric. Thus, if a single half-period (positive or negative half) is used as the truncated filter coefficient values, the phase response will not necessarily be linear. We propose the following scheme. First, extract a full period (positive and negative halves) of the sigma-delta truncator output and split the period into four quadrants. Next, consider each quadrant as either the first or last half of a symmetric linear phase FIR impulse response. The negative portions must be multiplied by minus one. Lastly, compute the mean-square error between each quadrant and the infinite precision FIR filter impulse response. The quadrant with the lowest mean-square error from the infinite precision case will generally have the best high frequency response characteristics.

To summarize, the proposed design procedure is outlined below:

1) Determine optimal infinite precision filter coefficients (impulse response) \( h_i(n) \).
2) Create a periodic extension of \( h_i(n) \) (5 periods is sufficient).
3) Scale the periodic extension using (2).
4) Pass the scaled and extended infinite precision input through the CSD sigma-delta truncator of figure 1b and extract the last complete cycle of the output.
5) Divide the extracted cycle into quadrants and use the truncated CSD values to create four candidate linear phase FIR filters.
6) Compute the mean-square error between each of the four filter coefficients and the original (scaled) infinite precision filter coefficients. Select the filter with the lowest mean-square error.
7) Compute the frequency response of the best filter to determine if the desired specifications have been met. If not, increase the number of taps or the number of non-zero CSD bits and iterate back through steps 1-6.

4 Example

As an example, we consider the filter suggested in example 1 of [2] for comparison. In this example, the filter impulse response is encoded with both the previous and the new CSD based sigma-delta schemes. The filter is designed using the Fourier method based on the transfer function:
\[
H(f) = \begin{cases} 
1.0 & f < 0.20 \\
(0.25 - f)0.05 & 0.2 \leq f \leq 0.25 \\
0 & f > 0.25 
\end{cases}
\]

Giving an impulse response of
\[h(n) = 1.25\text{sinc}^2(0.25n) - 0.8\text{sinc}^2(0.2n)\]

A 128 tap filter with infinite precision coefficients is determined using the window:
\[w(n) = 0.338946 + 0.481973\cos(\pi n/64) + 0.161054\cos(2\pi n/64) + 0.018127\cos(3\pi n/64)\]
\[n = -64, \ldots, 0, \ldots, 63\]

The infinite precision impulse response is shown in figure 5a. Figure 5b shows the result of oversampling this impulse response eight times (OSR=8) and passing the resulting sequence through the sigma-delta CSD truncator of figure 1b. Note the similarity between figures 5a and 5b even though the impulse response values in figure 5b are represented with only two non-zero bits. Figure 5c shows the result of oversampling by 8 and using the previously reported sigma-delta uniform truncator of figure 1a. Obviously, the CSD based approach provides a much closer approximation to the ideal impulse response.

As this oversampling rate (OSR) is increased, the truncated filter coefficients represent increasingly better approximations of the ideal frequency response but the filter must be operated at increasingly higher sample rates. Figure 7a shows the frequency response of the sigma-delta CSD truncated filter coefficients at various oversampling rates. As expected, the frequency response improves with increasing OSR. For comparison, figure 7b shows the frequency response of the sigma-delta uniform truncated filter coefficients at the same ORS values. As can be seen from figure 7, the sigma-delta CSD truncated filter has significantly improved filter characteristics (stopband attenuation) for a given OSR. For example, if a 50dB stopband attenuation is desired, the sigma-delta CSD truncated filter need only be oversampled by a factor of 8 whereas the previously reported sigma-delta uniform truncator requires an OSR of at least 512. Although both approaches require only a single addition per tap weight, the sample rate of the sigma-delta CSD truncated filter is reduced by 64 times.

5 Conclusions

A new oversampled sigma-delta differential encoding scheme was described which uses CSD code for the loop truncator. FIR filter coefficient multipliers can be reduced to a few adders. The implementation of the sigma-delta CSD truncation scheme was described in detail. A filter design example was presented and compared to a previously reported sigma-delta truncation scheme. The new scheme offered a 64 times improvement in the sample rate required to implement the filter without increasing the hardware complexity.

![Figure 5: Impulse response for Filter Example.](image-url)
(a) Ideal (no quantization)
(b) 2-bit CSD Sigma-Delta Truncation
(c) 2-bit Standard Sigma-Delta Truncation
Figure 6: Frequency Response for Filter Example for Various Oversampling Ratios
(a) 2-bit CSD Sigma-Delta Truncation
(b) 2-bit Standard Sigma-Delta Truncation

References


