# Components for Oversampled Signal Processors

John Monk
Faculty of Technology,
Open University,
Milton Keynes, MK7 6AA, UK
*j.monk@open.ac.uk*

***Abstract* – Oversampled converters trade transmission bandwidth for resolution. An idealized model gives insight into the signal encoding and thus how signals can be manipulated. Oversampling offers a form of signal processing that requires simple processing elements capable of exploiting the growing clock speeds available in integrated solutions and avoids the need for analog circuitry. This paper reviews common operations that can be performed on oversampled signals.**

## I. Introduction

Oversampled converters are widely used in consumer audio products[1] where they provide high resolution at low cost using low voltage supplies. Their advantages arise because they need only a single level comparator[2] and achieve fine resolutions by sampling beyond the normally assumed sampling frequency limit. Evolving higher logic speeds suggest feasible uses for oversampling techniques in IF and RF processing[3], but these applications require a wider repertoire of oversampled processing components.

In an oversampled converter, the analog signal is quantized to just two levels and thus the output power spectrum has a component representing the analog signal plus severe corruption attributable to the extreme quantization. Fortunately, converter designs distribute the energy of the quantization noise in frequency bands largely away from signal frequencies so that a significant proportion of quantization noise can be separated from the restored signal.

A simple feedback configuration can perform the conversion function but its form tends to obscure explanations. In this paper the configuration is transformed to give a simpler idealized open-loop arrangement which is behaviorally equivalent and which exposes the relationship between the converter's parameters and its performance.

Methods have been proposed, for example, for the addition of oversampled bit-streams by interleaving them, however this increases the required output channel bandwidth[4]. Other authors have used oversampled bit-streams in hybrid multiplication schemes[5] but the result demands an analog channel. The model shown here reveals how oversampled signals are encoded and how basic signal processing operations can generate binary data-streams without higher data rates or elaborate arithmetic.

## II. Oversampling

In the digital processing of analog signals, the analog signal is both quantized and sampled. Following Widrow's classic analysis[6], the quantization error is often characterized by noise with a rectangular probability density function that extends from an amplitude of zero up to the quantization interval $q$. The probability density is assumed to be uniform and therefore equal to $1/q$. It is also commonly assumed that the quantization errors are independent[7].

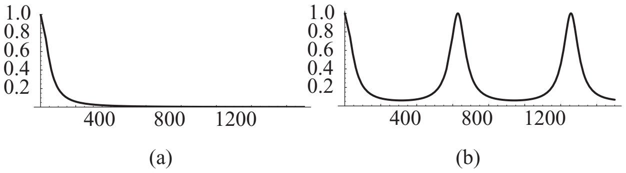

Fig. 1(a) A signal spectrum. (b) The sampled signal spectrum

Occasionally this assumption is invalid but usually the quantization errors can be modeled adequately by a white noise source with a total noise power of $q^2/12$.

Sampling introduces aliasing errors. The sampled signal spectrum repeatedly folds about multiples of half the sampling frequency. Fig. 1(a), for example, shows a signal spectrum and Fig. 1(b) shows the spectrum of the signal sampled at a rate of 100 samples per second and passed through a zero order hold. Errors occur because the tails of the folded spectra interfere with one another. Oversampling reduces the overlap and hence reduces aliasing errors.

Symmetry implies that the power distribution of sampled signals can be characterized by their spectrum over frequencies from zero to half the sampling frequency. Quantization errors are not band limited unless they are specially treated but, when folded and attributed to the lower frequency band, form a rectangular spectral distribution that accounts for the aggregate quantization power. Fig. 2(a) illustrates the folded spectrum of a sampled data signal with its broadband quantization noise included. Fig. 2(b) is the spectrum of the same signal sampled at a higher rate. The total quantization noise power is unchanged but the new sampling rate alters the frequency at which spectral folding takes place. The quantization noise power is therefore stretched over a wider band and consequently the spectral density of the quantization noise is lowered.

Sampling at a frequency of $2\Omega$ ensures that there is no aliasing of a signal which is band-limited in the frequency band $(0,\Omega)$. The total quantization error of $q^2/12$ spread uniformly over the band $(0,\Omega)$ gives a power spectral density of $q^2/(12\Omega)$. The spectral envelope of the signal is unaffected by doubling the sampling frequency to $4\Omega$ but the quantization noise is then spread over a bandwidth of $2\Omega$ and its spectral density reduced to $q^2/(24\Omega)$. The band-limited signal can be filtered by a low-pass filter. At higher

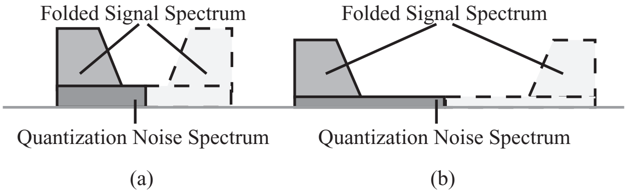

Fig. 2(a) A schematic sampled data spectrum with quantization noise. (b) The same spectra sampled at a higher rate.

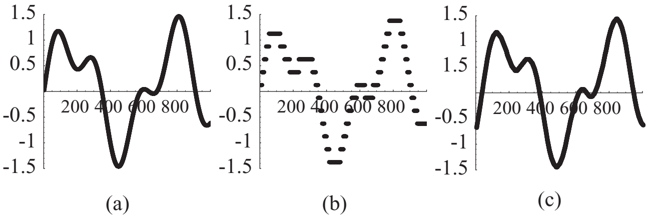

Fig. 3 (a) A band limited sampled waveform. (b) The sampled waveform quantized with a quantization interval of 0.25. (c) The quantized waveform filtered by a low-pass 64-sample moving average filter.

frequencies the broadband quantization noise will be attenuated and the total quantization noise reduced. For instance with a sample rate of $4\Omega$, quantization noise between $\Omega$ and $2\Omega$ can be removed without substantially interfering with the signal. The total quantization noise power, given by the integral of the product of the bandwidth of the noise times its spectral density, is then halved.

Fig. 3(a) shows a sampled signal with a bandwidth equivalent to a period of around 600 samples. Fig. 3 (b) is the same sampled signal quantized using a quantization interval of 0.25. Fig. 3(c) is the quantized signal filtered by a low-pass moving average filter averaging over 64 samples. It has introduced a delay of 32 samples but has visibly reduced the quantization noise

Although illustrated here by signals with power concentrated at low frequencies, the technique can be applied to any band-limited signal wherever that band limit is in the spectrum. Ultimately with sufficient over-sampling, a waveform can be adequately reconstructed from a quantized version quantized using just two quantization levels.

## III. Sigma-Delta Converters

Sigma-Delta converters are common devices for creating coarsely quantized oversampled signals. There are many configurations of Sigma-Delta converters[8]. Even the simplest configuration is difficult to analyze but following earlier work[9] the configuration can be transformed into an idealized equivalent that offers simpler explanations.

### *A. The basic converter.*

In the basic form of a Sigma-Delta converter shown Fig. 4(a), a feedback loop is formed around a filter, $\Phi$, a threshold device and a sampler. A key benefit arises from the simple quantizer that has a single threshold[10]. The output of the threshold device is 0 or 1 and the sampler provides a binary sampled output. The feedback path converts the digital output to analog levels and introduces a delay. Variations on this configuration use various filters and sometimes converters are cascaded. I consider a simple form with an integrator in the place of the filter $\Phi$.

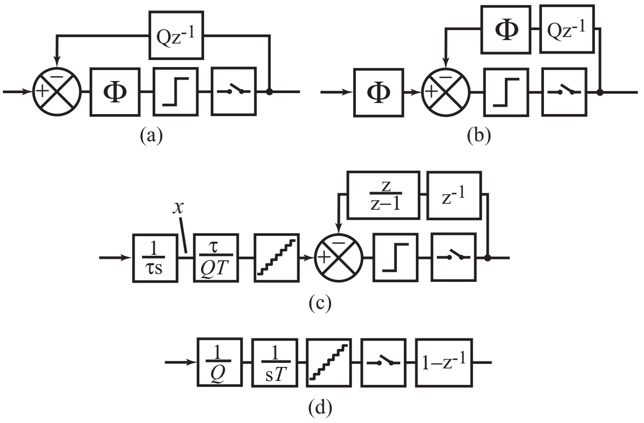

Fig. 4 Block diagrams of the transformation of a Sigma-Delta converter: (a) The basic converter. (b) The displacement of the filter $\Phi$. (c) The feedback loop moved to a digital section. (d) Idealized open-loop equivalent.

### *B. Transformation.*

The first transformation, shown in Fig. 4(b), replaces the integrator, $\Phi$, with two integrators that together have the same effect as the original in the forward path.

Assuming the logical outputs of 0 and 1 are translated into the corresponding analog levels of 0 and $Q$, nothing will be added by the integrator in the feedback loop when the output of the converter is 0. If the time constant of the integrator is $\tau$ and the period between samples is $T$, then a further $Q \times T/\tau$ will be added by the feedback integrator at the end of a sampling period when the converter output is a 1. The feedback path therefore, after integration, feeds back an integral number of quanta of magnitude $QT/\tau$. If the number of quanta is $m$ and the input integrator output is $x$ then the threshold device performs the test $x - mQT/\tau > 0$. This can be rearranged to give $x\tau/(QT) - m > 0$ and the outcome of the test generated by dealing entirely in integers and hence digitally. The test is $\mathrm{R}(x\tau/(QT) + \frac{1}{2}) - m > 0$, where the function R rounds its argument to the nearest integer and is equivalent to a multilevel quantizer with a quantization step of 1. The additional ½ can be absorbed in the quantizer design or in the initial conditions of the input integrator. Fig. 4(b) shows a revised configuration with a digital feedback integrator, the added multilevel quantizer and the additional gain of $\tau/(QT)$.

The threshold device can be made redundant by choosing operating conditions that restrict the threshold device's input, $\mathrm{R}(x\tau/(QT) + \frac{1}{2}) - m$, to either 0 or 1. In these circumstances, the feedback loop, now in the digital domain and without the threshold device, can be replaced by an equivalent derived using the classic result:

$$\frac{G}{1+GH} = \frac{1}{1+z^{-1}\left(z/(z-1)\right)} = \frac{z-1}{z} = 1 - z^{-1}. \qquad (1)$$

This transfer function forms the difference of successive samples and gives an indication of the rate of change of the output of the quantizer. The Sigma-Delta converter is therefore equivalent to the idealized configuration of Fig. 4(d).

### *C. Operational bounds.*

In the converter shown in Fig. 4(d), a zero is output from the converter when there is no change in the quantizer output between adjacent samples. If there is a rise of one quantization interval over one sample interval then the difference is 1 and the converter output is a 1. A larger rate of change however will generate an output that cannot be represented in a single bit and exceeds the operating restrictions that led to the elimination of the threshold device. This constrains the rate of change at the input to the quantizer. For an input to the converter of amplitude $A$, the input to the quantizer changes by $T \times A/(QT)$ in one sample interval. This change must not exceed one quantization interval, hence $A \le Q$. Similarly, if the rate of change recorded by the digitizer is negative then the result cannot be accommodated in the binary output channel. The bounds on the converter input are therefore $0 \le A \le Q$ and the gain provided by the converter with an output in the range 0 to 1 is $1/Q$. This bound is impractical since it implies that the integrator in Fig. 4(d) has a monotonically increasing output. However, Fig. 4(d) is an idealization thus the impractical bound is acceptable provided care is taken in interpreting the internal behavior of the idealized configuration.

As it stands the model does not permit the input of bipolar signals. Restricting the input to $\pm Q/2$ and adding a bias, as shown in Fig. 5, accommodates this.

### *D. Reconstruction of the analog signal.*

The output of the Sigma-Delta converter is a bit-stream operating at a relatively high bit-rate with a high level of high frequency quantization noise. A multilevel digital signal can be extracted by passing the bit-stream through a low pass digital filter or if an analog output is required an analog or hybrid low pass filter to attenuate folded spectra as well as quantization noise.

## IV. Noise shaping

The model shows that the Sigma-Delta converter effectively biases and filters the incoming signal before quantization and provides further digital filtering after quantization. The combined effects of the input and the output filters restore the signal components in the output but only the output filters affect the quantization noise. Fig. 6 illustrates the changes in spectra through a converter followed by a post-processing digital filter, Γ(z).

The input integrator emphasises the low frequencies in the signal. The quantizer adds quantization noise with a uniform broadband spectrum and the combination of signal and noise pass through a differencing element that restores the signal spectrum but exaggerates the high frequency components of the noise and reduces the low frequency components. Finally, a low-pass filter removes power in the frequencies dominated by the noise and hence reduces the effects of quantization.

Overall the converter restores the input signal but adds broadband noise that is shaped by output filters. With a signal in the band $(0, \Omega)$ and an oversampling ratio of $k$, the sampling frequency is $2k\Omega$. The power spectral density of the quantization noise generated by the quantizer distributed over the band $(0, k\Omega)$ and referred to the input levels is therefore $Q^2/(12k\Omega)$. After differencing this becomes

$$\frac{Q^2}{12k\Omega}\left|1-z^{-1}\right|^2_{z\to e^{j\omega T}} = \frac{Q^2}{12k\Omega}\left(2\sin\frac{\omega T}{2}\right)^2 . \qquad (2)$$

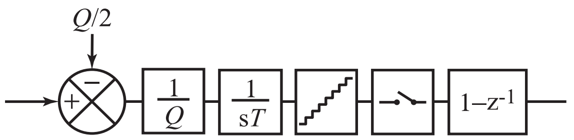

Fig. 5 Amended model with the input bias added for bipolar operation.

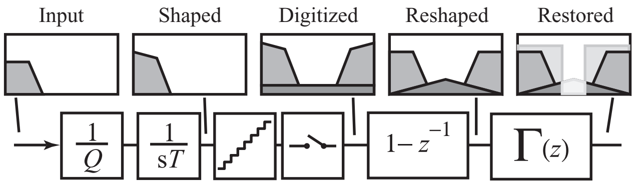

Fig. 6 The transformation of the signal and quantization noise spectra.

If the low pass output filter is ideal this quantization noise will be wholly attenuated outside the band $(0, \Omega)$, and the total noise power reduced to

$$\frac{Q^2}{12k\Omega}\int_0^{\Omega}\left(2\sin\frac{\omega T}{2}\right)^2 \mathrm{d}\,\omega = \frac{Q^2}{6k\Omega T}\left(\Omega T - \sin\Omega T\right). \qquad (3)$$

$T$ is $1/4\pi k\Omega$ hence $\Omega T$ is $1/4\pi k$ and is small because the oversampling ratio $k$ will be moderately large, therefore using a series expansion the noise power is approximately

$$\frac{Q^2}{6k\Omega T}\left(\frac{(\Omega T)^3}{6}-\frac{(\Omega T)^5}{120}+\cdots\right) \approx \frac{Q^2\Omega^2T^2}{36k} = \frac{Q^2}{576\pi^2k^3} . \qquad (4)$$

The bound on the input signal is $\pm Q/2$. The maximum signal power is $Q^2/4$ giving a signal to noise ratio of $144\pi^2k^3$. In decibels this is approximately $31.5 + 30\log_{10}k$.

## V. The Processing elements

### *A. The signals*

In the oversampled converter, an analog input is converted to a bit-stream. The analog input can be restored, subject to the inexactness introduced by sampling and quantizing, by first obtaining a data-stream which is a scaled and biased version of the bit-stream. This data-stream carries the properly scaled spectral components of the analog waveform that can be extracted by filtering out the quantization noise and folded spectral components.

Arithmetically, bit-streams are scaled and biased versions of the data-streams that take values from the set $\{-Q/2, Q/2\}$. If the represented data-stream is $a_n$ then its representative bit-stream is $A_n = (a_n + Q/2)/Q$ and this implies that the data-stream can be obtained from the bit-stream by calculating $a_n = Q(A_n - ½)$.

### *B. An adder*

From an analog perspective, a bit-stream with amplitude values $\{0, 1\}$ incorporates a bias of ½. The effect of adding two bit-streams is not only to create a sum but also to double the bias. This excess bias must be removed to provide a representative sum. When two bit-streams $A_n$ and $B_n$ represent data-streams then each data-stream, $a_n$ and $b_n$, can be derived from its related bit-stream by the equations

$$a_n = Q\left(A_n - ½\right) \quad b_n = Q\left(B_n - ½\right). \qquad (5)$$

The result should represent a data-stream $y_n = a_n + b_n$ and this is represented by a bit-stream, $Y_n$, where

$$Y_n = \frac{1}{Q}\left(y_n + Q/2\right) = \frac{1}{Q}\left(\left(a_n + b_n\right) + Q/2\right). \qquad (6)$$

Substituting the expressions for the data-streams into the expression for the sum gives $Y_n = A_n + B_n - ½$. There are two problems with this result. Firstly, adding two binary streams directly gives a result that can take on the values in the set {0,1,2} given by the possible sums

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=2\,.$$

The ternary results cannot be borne by a binary output stream. Secondly, the bias integrated into the two bit-streams takes the result outside of the domain of integers.

One way of dealing with this is to output a 0 or a 1 and to carry forward any inconvenient residue from one bit interval to the next. The process involves forming a sum of the two bit-streams with any earlier residue and then subtracting the surplus bias of one half. If the residue carried forward at sample instant $n$ is $R_n$, the aggregate is $\Sigma_n = A_n + B_n + R_{n-1} - ½$. When this aggregate is greater than zero, a 1 can be output and deducted from the aggregate. Any residue is carried forward. When the aggregate is zero or less, it is carried forward unaltered and the output receives a 0. So for an output bit-stream $D_n$ the algorithm is

$$\Sigma_n = A_n + B_n - ½ + R_{n-1} \quad D_n = \begin{cases} 0 & \Sigma_n < 0 \\ 1 & \Sigma_n \geq 0 \end{cases} \quad R_n = \Sigma_n - D_n \tag{7}$$

Alternatively the last step in this algorithm can be incorporated into the first stage of the following calculation, the residue can also be scaled without affecting the result and because the output stream is binary, the negation of the stream $D$ can be expressed in terms of the complement of the output, $\overline{D}$ . Thus the algorithm becomes

$$\Sigma_n = 2A_n + 2B_n - 3 + \Sigma_{n-1} + 2\overline{D}_{n-1} \quad D_n = \begin{cases} 0 & \Sigma_n < 0 \\ 1 & \Sigma_n \geq 0 \end{cases}. \tag{8}$$

The block diagram in Fig. 7(a) shows a configuration that will perform this task. The adder is an elaborate component but this can be reduced in complexity since the inputs are bit-streams Also the awkward bias terms are constant so that their subtraction can be dealt with by integrating their effects into the structure of the adder. Only elementary small valued integer arithmetic is required.

This configuration is similar in form to the core of a Sigma-Delta converter though in this case the components operate entirely in the digital domain with the residue accumulator performing a digital integration function and with negative feedback provided from the binary output of a threshold device. The adder design can be reduced to the diagram of Fig. 7(b) with an operator Θ, in this case addition, preceding the conversion of a multilevel digital signal to a binary stream.

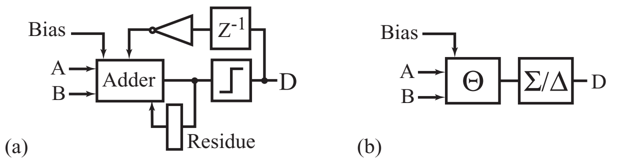

Fig. 7 (a) A configuration that performs addition on bit-streams and generates a bit-stream output. (b) An equivalent simplified diagram.

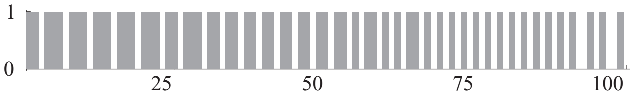

Fig. 8 The binary input representing a portion of a sinusoid.

As an example, a sinusoid with a period of approximately 314 samples and its second harmonic were converted to bit-streams. A portion of the bit-stream of the higher frequency signal is shown in Fig. 8. The two bit-streams were added in a bit-stream adder and the binary result filtered by a 64-sample moving average low pass filter. The outcome is shown in Fig. 9 (a) with the horizontal axis indicating the time as a multiple of the sample interval. The spectral components of the result are shown in Fig. 9(b), which reveals the two major components and negligible levels of quantization noise

*C. A scaler*

Another common signal processing operation involves scaling a data-stream. For a data-stream $a_n$, scaling by a factor $m$ should produce a data-stream $ma_n$. When $ma_n$ is represented by the bit-stream $Y_n$, its scaled version is represented by a sequence of the form

$$Y_n = \frac{1}{Q}\left(ma_n + Q/2\right). \tag{9}$$

Substituting the expression for $a_n$, which relates it to the bit-stream $A_n$, gives the result

$$Y_n = \frac{1}{Q}\left(mQ\left(A_n - ½\right) + Q/2\right) = m\left(A_n - ½\right) + ½\,. \tag{10}$$

Again the calculated values will not be restricted to those in the set {0, 1} but once more the technique of carrying forward any residue from a calculation can be employed to create a binary stream. In this case the algorithm is

$$\Sigma_n = 2mA_n - m - 1 + \Sigma_{n-1} + 2\overline{D}_{n-1} \quad D_n = \begin{cases} 0 & \Sigma_n < 0 \\ 1 & \Sigma_n \geq 0 \end{cases}. \tag{11}$$

For $m = 2$ for example the recurrence relation becomes

$$\Sigma_n = 4A_n - 3 + \Sigma_{n-1} + 2\overline{D}_n\,. \tag{12}$$

Fractional values of $m$ are awkward, if integer arithmetic is preferred. However, the residues can be rescaled to restore integer factors in the calculation so for example, for $m$=½ rescaling gives the recurrence relation

$$\Sigma_n = 2A_n + \Sigma_{n-1} + 4\overline{D}_{n-1} - 3\,. \tag{13}$$

*D. Operations on a single bit-stream*

Interchanging positive and negative values negates a data-stream. For the representative bit-streams this simply

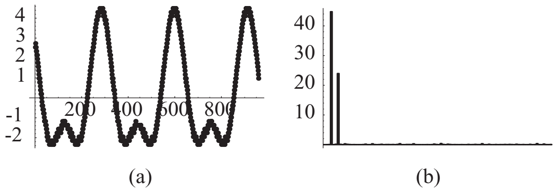

Fig. 9 (a) The sum of two sinusoids reconstructed from a bit-stream using a moving average filter over 64 samples. (b) The power spectrum of showing the power in the two added sinusoids and the low quantization noise.

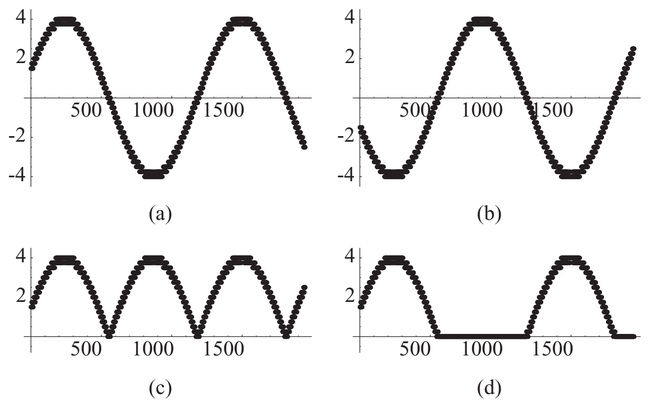

Fig. 10 (a) A reconstructed sinusoid. (b) The reconstructed sinusoid after logically inverting the bit-stream. (c) The result of applying an inverted exclusive OR operation to adjacent samples. (d) The result of applying an AND operation on adjacent samples.

means inverting the ones and zeros, consequently the logical inverse of a bit-stream represents the arithmetic inverse of a data-stream. Arithmetic negation therefore requires a single logic inverter, and subtraction can be handled by adding a logically inverted bit-stream.

Bit-streams can be translated into a ternary representation by forming the sum between consecutive bit-stream values. If this result is biased and scaled then the ternary data-stream has the values {-*Q*/2,0,*Q*/2} with the negative values contributing to negative going parts of the represented waveform. If the negative values are made positive then the negative going parts of the waveform are inverted and the waveform is rectified. The combined operation of adding, scaling and biasing is equivalent to performing the logical negation of the exclusive-OR operation on consecutive bit-stream samples. In the rectification process the ternary values represent data-streams directly without bias. An additional bias term must be included to convert the result to a binary stream consistent with other biased bit-streams. An algorithm for rectification and restoring the bias level is

$$\Sigma_n = \overline{A_n \oplus A_{n-1}} + \Sigma_{n-1} + 2\overline{D}_n + 1 \quad D_n = \begin{cases} 0 & \Sigma_n < 0 \\ 1 & \Sigma_n \geq 0 \end{cases}. \qquad (14)$$

Half-wave rectification is obtained by setting negative ternary values to zero. This is equivalent to an algorithm that forms a logical AND on neighboring bit-stream values:

$$\Sigma_n = A_n \wedge A_{n-1} + \Sigma_{n-1} + 1 + \overline{D}_n \quad D_n = \begin{cases} 0 & \Sigma_n < 0 \\ 1 & \Sigma_n \geq 0 \end{cases}. \qquad (15)$$

Fig. 10(a) shows a reconstructed waveform, Fig. 10(b) shows the reconstructed waveform inverted by logically inverting its representative bit-stream. Fig. 10(c) shows the full-wave rectified waveform generated by processing the bit-stream with an exclusive-OR and Fig. 10(d) shows the half-wave rectified sinusoid after the bit-stream has been processed using an AND operation (The NOR operation reproduces the alternate half cycles).

### *E. Multiplication*

The scaled and biased product of two data-streams corresponds to the representative bit-stream

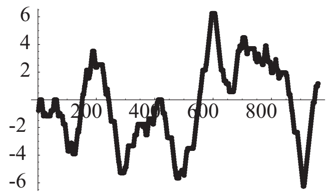

Fig. 11 The reconstructed output of a simple bit-stream multiplier.

$$\frac{1}{Q}\left((a_n b_n) + \frac{Q}{2}\right) = \frac{1}{Q}\left(\left(Q(A_n - ½) \times Q(B_n - ½)\right) + Q/2\right)$$
$$= Q\left((A_n \times B_n) - ½(A_n + B_n) + ¼\right) + ½ \ . \qquad (16)$$

To obtain a correctly biased result an additional ¼ must be added and the spurious linear term in $A_n$ and $B_n$ subtracted. Scaling the residue creates the recurrence relation

$$\Sigma_n = Q\left(4A_n B_n - 2(A_n + B_n) + 1\right) - 2 + \Sigma_{n-1} + 4\overline{D}_{n-1}. \qquad (17)$$

This calculation is relatively simple, since $A_n$ and $B_n$ are bit-streams. The terms involving the input streams result in only two values — 0 or –2 — and mimic the truth table for the exclusive-OR. The realization of such a multiplier is therefore straightforward. However, bit-streams incorporate substantial quantization noise in addition to the represented data. This noise is normally concentrated at high frequencies and is removed by filtering. Unfortunately, multiplication generates intermodulation products that may contribute substantial noise components at the signal frequencies. Fig. 11 shows, for example, the corrupted result of multiplying two bit-streams representing sinusoids after reconstruction. The results are therefore not always satisfactory.

Incorporating filters at the input of the multiplier can reduce unwanted frequency components and reduce the degree of intermodulation. An algorithm that filters an input stream by forming a moving average over four samples and before calculating the square of the smoothed result is

$$8 + \frac{1}{Q}(A_n + A_{n-1} + A_{n-2} + A_{n-3} - 2)^2 + \Sigma_{n-1} + 16\left(\overline{D}_{n-1} - 1\right) (18)$$

This algorithm does not involve complicated multipliers since the bit-streams are all binary and cross products can be provided by AND gates. Fig. 12(a) is the output of a configuration that executes the squaring algorithm operating on a sinusoid with a frequency equivalent to about 628 samples. The algorithm averaged over 16 samples. The output exhibits the frequency doubling that would be expected but, as shown in the spectrum of Fig. 12(b), the multiplier introduces negligible intermodulation products in addition to the anticipated zero frequency term and the single sinusoidal coefficient.

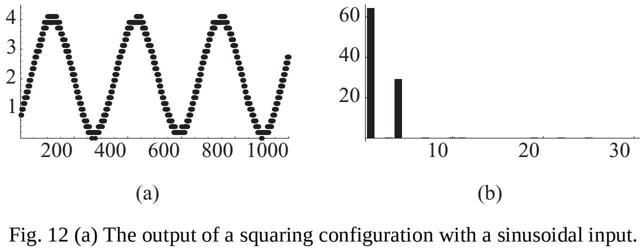

Fig. 12 (a) The output of a squaring configuration with a sinusoidal input. (b) The spectral components of a reconstructed squared sinusoid output from a squaring configuration incorporating an input filter.

### *F. A generalization*

Operations can often be combined into single simple processing elements. For example, several bit-streams can be scaled individually then added in a single calculation. Suppose the data-streams $a_n$, $b_n$ and $c_n$ to form the weighted sum $y_n = m_a a_n + m_b b_n + m_c c_n$. Substituting expressions involving the related bit-streams into the expression for the output data-stream creates the recurrence relation

$$\Sigma_n = 2(m_a A_n + m_b B_n + m_c C_n) \\ -1-(m_a + m_b + m_c) + \Sigma_{n-1} + 2\overline{D}_{n-1}. \qquad (19)$$

This calculation offers the prototype for the construction of FIR filters using delayed bit-streams as the input sequences. There are three approaches to filter design:

1. Filter the incoming bit-stream, obtain a multilevel output and convert this to a bit-stream using a Sigma-Delta converter;

2. Incorporate the filter in the Sigma-Delta modulator;

3. Create filter building blocks such as integrators from Sigma-Delta modulators.

The various options introduce different levels of quantization noise and complexity. The main circuit advantages arising where single bit addition or multiplications take place[11]. As an example a first order filter integrated with a Sigma-Delta modulator creates the recurrence relation

$$\Sigma_n = A_n - D_{n-2} + 2\Sigma_{n-1} - \Sigma_{n-2} + N(D_{n-2} - D_{n-1}), \qquad (20)$$

where $N$ is the approximate time constant of the filter measured in sample intervals. Note that some of the arithmetic is carried out on raw bit-streams and can be performed economically. A substantial simplification can also be made if $N$ is a power of two. Fig. 13 shows a block diagram of the filter arranged to show an inner loop that forms a conventional Sigma-Delta converter but using integer arithmetic. Fig. 14(a) is the reconstructed output waveform of such a filter with an input bit-stream representing a periodic input signal composed of ten harmonics. The filter time constant was equivalent to roughly 256 samples. Fig. 14(b) shows the spectral components of the reconstructed input and output bit-streams. The first order filter characteristics are clearly exposed in the darker bars representing the output spectrum.

## VI. Conclusions

Increasing frequencies of circuit operation widen the field for the application of oversampled techniques. This demands a broader variety of signal processing operations and fortunately there are readily analyzed, proven ranges of signal processing operations that can be performed on bit-streams. Bit-stream processing enables the complexity of signal processing solutions to be reduced in return for higher speeds of operation and in addition offers designers the well-known benefits of oversampled converters that avoid the need for analog component matching and permit circuit operation at low voltages.

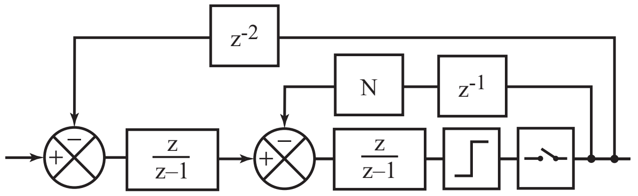

Fig. 13 A block diagram of the first order filter.

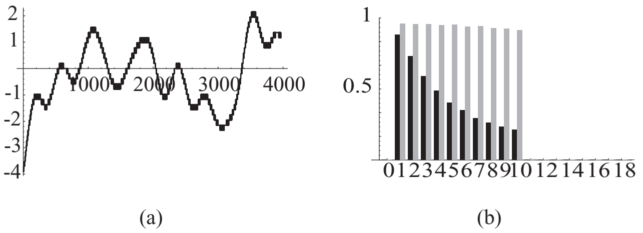

Fig. 14 (a) The restored output of the first order filter. (b) The spectral components of the restored filter input and output signals.

## VII. References

[1] S. Rabii and B.A. Wooley, "A 1.8-V Digital-Audio Sigma-Delta Modulator in 08μm CMOS," *IEEE JSSC*, vol.32, no.6, June 1997, pp.783–796.

[2] Kun Lin, Kan Zhao, E. Chui, A. Krone and J.Nohrden, "Digital Filters for High Performance Audio Delta-sigma Analog-to-digital and Digital-to-analog Conversions," *Proceedings of the 1996 International Conference on Signal Processing*, pp.59–63.

[3] A.R. Feldman, B.E. Boser and P.R. Gray, "A 13-bit, 1.4MS/s Sigma-Delta Modulator for RF Baseband Applications," *IEEE JSSC*, Oct 1998, vol.33, no.10, pp.1462–1469.

[4] F. Maloberti and P. O'Leary, "Processing of Signals in their Oversampled Delta-Sigma Domain," in *Proceedings of the 1991 International Conference on Circuits and Systems*, Shenzhen, China, pp.438–441.

[5] E. Dallago, G. Sassone, M. Storti and G. Venchi, "Experimental Analysis and Comparison on a Power Factor Controller Including a Delta-Sigma Processing Stage," *IEEE Trans on Industrial Electronics*, vol.45, no.4, August 1998, pp.544–551.

[6] B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," *IRE Trans. On Circuit Theory*, CT-3, 4, 1956, pp.266–276,

[7] G.F. Franklin and J.D Powell, *Digital Control of Dynamic Systems*, Addison Wesley, London: 1980, p.192.

[8] J.C. Candy and G.C. Temes, (eds), *Oversampling Methods for A/D and D/A Conversion*, IEEE Press: 1992.

[9] J.E. Flood and M.O.J. Hawksford, "Exact Model for Delta Modulation Processes," *Proc. IEE*, vol.118, 1971, pp.1155–1161.

[10] B.E. Boser and B.A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE JSSC*, vol.23, no.6, Dec. 1988, pp.1298–1308.

[11] D.A. Johns and D.M. Lewis, "Design and Analysis of Delta-Sigma Based IIR Filters," *IEEE Transactions on Circuits and Systems*, vol.40, no.4, April 1993, pp.233–240.