DSP for analog engineers

Most DSP applications require some analog or RF support circuitry. Having an understanding of the DSP portion of the system is useful to the analog designer.

By Mark Kolber

The most simple digitial signal processing (DSP) system consists of only an analog-to-digital (A/D) converter and a digital-to-analog (D/A) converter. However, some of the signal processing that can take place in the digital domain must be considered. (See Figure 1.)

A/D conversion

Analog signals are continuous in time and continuous in amplitude, existing at every instant of time with any possible intermediate amplitude value. DSPs mathematically process signals represented as a series of numbers that are discrete in time and discrete in amplitude. A/D conversion consists of *sampling, hold, quantize* and *code.* Sometimes these four functions are integrated into one device. (See Figure 2.)

Sampling

Sampling converts a signal that is *continuous in time* to one that is *discrete in time.* The number of samples taken per-second is called the *sampling rate* or *sampling frequency Fs.* The Nyquist theorem says that, if sampling is performed at a rate exceeding the Nyquist rate, or 2 times the bandwidth of the signal, no signal information is lost. Sampling can be thought of as a re-mapping of the frequency domain. The input frequencies exist in a line that extends from zero to infinity. The line is rolled up into a circle where the sampling rate Fs, which is normalized to 2pi, is mapped back to 0 Hz or DC.

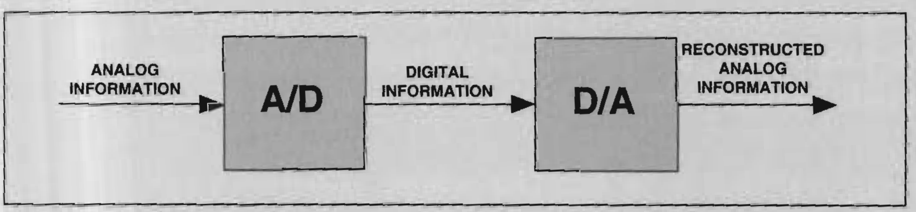

Figure 1. A simple DSP system.

Audio signals recorded on compact discs (CDs) extend from 20 Hz to 20 kHz and are sampled at Fs = 44.1kHz. Telephone channels with voice components between 300 Hz and 3 kHz are usually sampled at Fs = 8 kHz.

Similarly, the upper bandwidth limit Fs/2, called the *folding frequency* or *Nyquist limit*, that is normalized to pi, is mapped to Fs/2. After sampling, no frequency can exist above Fs/2. Any input signal component that exceeds the folding frequency, Fs/2, will be *aliased* or *folded back* and will reappear below the folding frequency. For example, if Fs = 8 kHz, the folding frequency is 4 kHz. An input signal component at 7 kHz, which is 3 kHz above the folding frequency, will be aliased to 3 kHz below the folding frequency and will reappear as a 1 kHz signal. Note, when an input signal between Fs/2 and Fs is aliased back to the region between 0 and Fs/2, it appears spectrally inverted just as if it had been frequency-translated using a mixer with high side injection. (See Figure 3.)

The Z transform converts between the time domain and this rolled up frequency domain in a way analogous to the way that the fourier transform con-

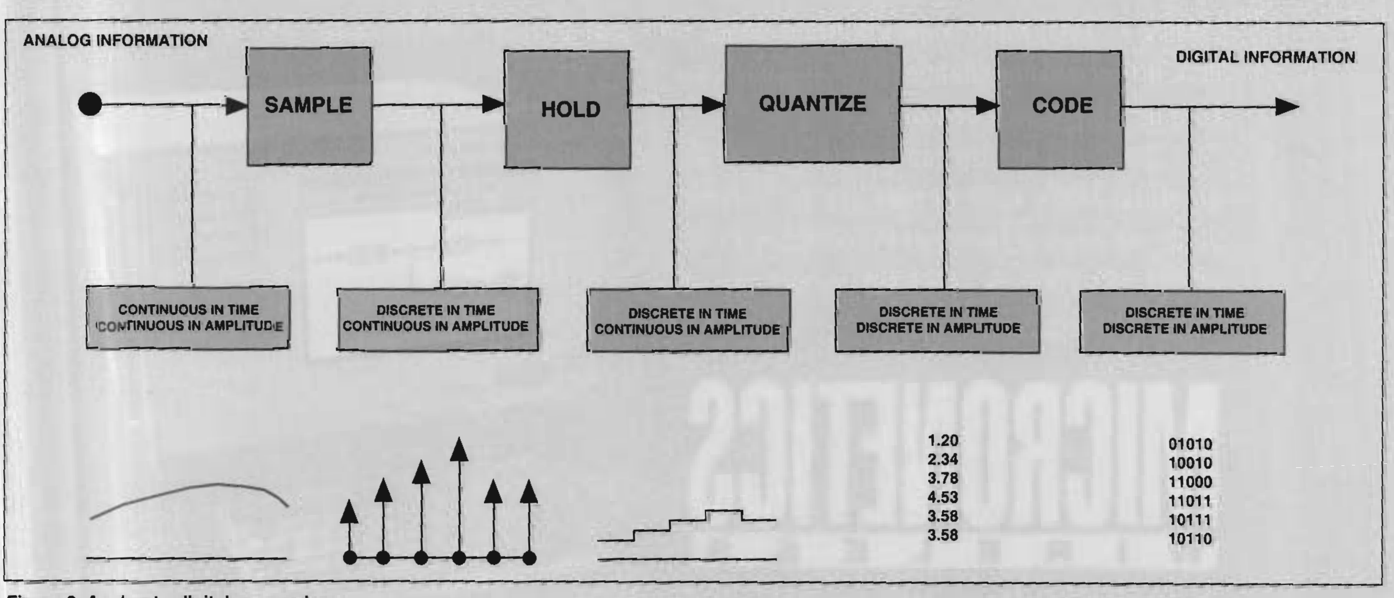

Figure 2. Analog-to-digital conversion.

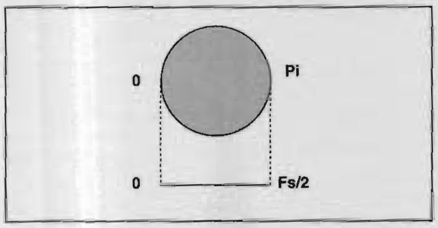

Figure 3.

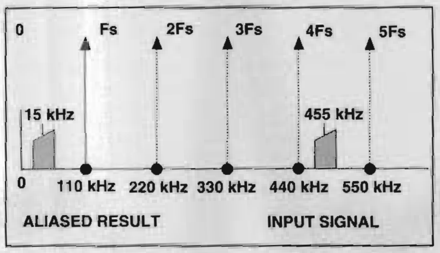

Figure 4. Subsampling.

verts between the time domain and the normal straight-line frequency domain.

Anti-alias lowpass filters are used to attenuate input signal components above the folding frequency before sampling. The design must consider the transition bandwidth and finite rejection of the filter, as well as any noise and distortion products above Fs/2 that may be re-generated after the filter because these will all be aliased and will reappear below the folding frequency.

Subsampling is a useful form of aliasing used to downconvert and sample bandpass signals that are centered above Fs/2. The Nyquist criterion is not violated, and no information is lost as long as Fs/2 exceeds the bandwidth of the input signal. For example, a 20 kHz-wide IF signal centered on 455 kHz can be subsampled using Fs = 110 kHz. The 4th harmonic of Fs, 440 kHz, will alias the 455 kHz center frequency down to 15 kHz. A bandpass, rather than a lowpass filter, is used to limit the bandwidth of the input signal to prevent unwanted aliasing. Depending on the relationship between the bandwidth and the center frequency of the input signal, the minimum usable sampling rate will lie between 2B and 4B where B is the bandwidth of the signal. (See Figure 4.)

Ideally, sampling is performed at exactly equally spaced time intervals. Any unwanted variation in the time interval is called *aperture jitter*. This is equivalent to phase noise on the sampling frequency, which will be imparted to the sampled signal. In many ways, sampling is analogous to frequency conversion using a mixer or multiplier. Sampling multiplies the input signal by a series of impulses. The resulting frequency translation is analogous to using a mixer with an LO that contains harmonics. Conversions that use harmonics of the LO are analogous to aliasing or subsampling.

Hold and quantize

After sampling, the signal is discrete in time but it is still continuous in amplitude. The signal is made discrete in amplitude as well by quantizing or representing each sample as a number with finite resolution. For example, this can be accomplished with a series of comparators. A hold function may be used to maintain the sampled value constant during the time it takes to quantize the signal.

Some of the techniques for quantizing include: *successive approximation, integration, flash* and *iterative*. Successive approximation converters use a D/A converter, comparator and control logic to perform a binary search for the digital value that is as close as possible to the input value. Integration-type converters use a ramp and a comparator to start and stop a digital counter whose output represents the value of the sample. These are relatively slow techniques, so these converters are primarily used for data rather than signal acquisition. Flash converters use a bank of comparators and voltage references, one for each quantization level. This converter is fast and is often used for signal acquisition. However, the number of comparators required rises exponentially with the number of output bits. Iterative converters use a smaller flash converter in a multistep sequence that results in a compromise between speed and cost.

Unlike Nyquist sampling, which loses no information, quantizing produces an error called *quantizing noise*. Because the hardware cannot process an infinite number of bits, the amplitude of the signal cannot be represented with infinite resolution. The difference between the exact sample value and the quantized value is quantizing error. Quantizing noise can usually be approximated as white noise if the input signal is continuously varying, is large compared to the difference in quantizing levels, and does not contain frequency components that are related to the sampling rate. *Dithering*, or adding a known signal to the input, is sometimes used to ensure the white noise approximation is valid. The dithering signal can be subtracted or filtered out in the digital domain. Under these conditions, the signal-to-quantizing noise ratio (SNR) is approximately:

$$\mathrm{SNR(dB)} = (6.02 \cdot n) + 1.25$$

and the number of possible signal amplitude levels is:

$$L = \left(2^n\right) - 1$$

where:

SNR = the signal to quantizing noise ratio in dB

n = the number of binary bits used to represent the levels

l = the number of possible quantization levels

Increasing the binary word size by one bit doubles the number of quantization levels and improves the SNR by about 6 dB. Audio CDs use 16 bits, providing an SNR of about 98 dB. The dynamic noise and linearity performance of practical A/D converters operating at RF frequencies are often specified by the effective-number-of-bits (ENOB) available, which is usually less than the actual number of output bits.

Oversampling is a technique that uses a sampling rate higher than the Nyquist rate. Because quantizing noise is spread over the full spectrum from 0 to Fs/2, if the signal occupies only a small portion of this range, much of the quantizing noise can be filtered out using DSP filters. Thus, over-sampling allows for the tradeoff of sampling rate vs. resolution. *Sigma delta modulation* is another technique for reducing the effects of quantization. It quantizes the changes in a signal between samples rather than the signal itself. These two techniques are combined in a sigma delta converter that uses a very high speed 1-bit sigma delta A/D and digital filtering to create a converter with excellent linearity.

A form of logarithmic quantizing often used for telephony called *Mu-law coding* uses non-linear quantizing where the quantized number is proportional to the logarithm of the signal. The quantizing steps effectively represent decibel values rather than volts. This causes the quantizing noise to be larger when the signal is large and smaller when it is small. Because the log of 0 is negative infinity, practical Mu-law converters must deviate from a

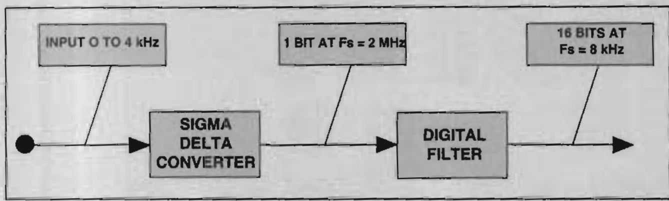

Figure 5. Sigma delta conversion.

DECIMAL NUMBER	FRACTIONAL NUMBER	STRAIGHT BINARY	OFFSET BINARY	2'S COMPLEMENT	SIGN & MAGNITUDE
+7	+7/4	111			
+6	+6/4	110			
+5	+5/4	101			
+4	+4/4	100			
+3	+3/4	011	111	011	011
+2	+2/4	010	110	010	010
+1	+1/4	001	101	001	001
0	0	000	100	000	000 or 100
–1	–1/4		011	111	101
–2	–2/4		010	110	110
–3	–3/4		001	101	111
–4	–4/4		000	100	

Table 1. Examples of popular codes used for bipolar signals for a 3-bit word size.

true logarithmic transfer function below some small signal level. The exact nature of this deviation is defined by the particular Mu-law curve being used. Because of its non-linear characteristic, this type of converter is not usually used for DSP applications. (See Figure 5.)

Coding

Coding is the representation of the quantized values in a particular numerical format usually a form of binary. Straight binary is the natural choice for unipolar signals. The various codes used for bipolar signals differ in their representation of the sign and the transition from positive to negative values. Examples of popular codes are shown in Table 1 for a 3-bit word size.

Two's complement is often used to represent bipolar signals because it is a form of modulo arithmetic which simplifies the addition and subtraction of a series of numbers. As long as the final result is in range, the sum will be correct in spite of the fact that some of the intermediate results may have been out of range. Multiplication, however, is easier using sign-magnitude representation.

Within a system, the A/D converter, numerical processing section and D/A converter must all use the same method of representation or the numbers must be converted. Offset binary and two's complement are popular for DSP. Notice that converting between offset binary and two's complement simply requires inverting the most significant bit.

Dynamic range considerations

The analog input signal must be properly scaled to fully utilize the dynamic range of the A/D converter. If the input signal is too small, the quantizing SNR will suffer. If the signal is too large, clipping may occur. Although most analog components have soft compression characteristics, exceeding the upper limit of an A/D converter produces hard clipping, which is not "well-behaved." The resulting high-order distortion increases rapidly rather than gradually as more clipping occurs. Non-linearities within the dynamic range of the A/D converter can also create high-order distortion products.

When the analog input contains a large number of signals, such as in a multichannel communications system, the dynamic range of the A/D converter must be carefully considered. The dynamic range of the A/D converter must accommodate not only the range of the desired channel but also the total of all the undesired channels as well. For this reason, many system designs include some analog pre-filtering to reduce the number of channels and hence dynamic range applied to the input of the A/D converter.

The driving impedance presented to the input of the A/D converter must also be considered. The various comparators within the A/D converter can cause its input impedance to vary as a function of the input signal. The converter must be driven by the specified source impedance, usually 50 Ω, in order to obtain the specified linearity performance.

After the signal has been converted to the digital domain, there are additional dynamic range considerations. The sum of 2 N bit numbers require N+1 bits, and the product of 2 N bit numbers require 2N bits. To limit the number of bits needed within a system and also to prevent overflow, the least significant bits of results are usually rounded or truncated. This increases the amount of quantizing noise but prevents numerical clipping. Truncation is easier to implement but can cause a small bias in the errors, which may accumulate. If the possibility of numerical overflow exists within the system, a clip function should be included to maintain the overflowed value at full scale rather than allowing it to wrap back around to an indeterminate smaller value.

Floating point binary representation can be used to increase the dynamic range available in the digital domain. With fixed point representation, the smallest value that can be represented is one least significant bit (LSB). The largest is 2(N – 1) times larger where n is the number of binary bits in the word. Floating point representation uses two words, one representing the mantissa M and the second an exponent or characteristic C. The value x is represented as:

$$x = M \times 2^{c}$$

M can be signed to represent both positive and negative numbers and is usually normalized between 1/2 and 1. C can be signed to represent values greater, or less than, one. For a given number of bits, floating point is able to represent a much larger dynamic range compared to fixed point. The resolution of a floating point is not fixed but is proportional to the value of the input signal. The sum of two floating point numbers is the sum of the mantissas if the exponents are equal. The product of two floating point numbers is found by multiplying the mantissas and adding the exponents. Floating point arithmetic produces roundoff errors for both addition and multiplication; fixed point produces roundoff errors only for multiplication. Because of the wide dynamic range, floating point reduces the occurrence of overflow errors. The Institute of Electrical and Electronic Engineers (IEEE) standard 754 defines the details of a commonly used floating point representation.

Digital-to-analog conversion

In the simple DSP system, the digital output of the A/D converter is connected directly to the input of the D/A converter. An ideal D/A converter will generate an output voltage that exists for only an instant in time for each sample. The output value between samples must be interpolated by the reconstruction filter, which will make the analog output again continuous in both time and amplitude. If the sampling meets the Nyquist criteria, and if the reconstruction filter eliminates all frequencies above the folding frequency, the interpolation causes no error and the output will be an exact replica of the original signal, except for the quantizing errors.

The use of very narrow pulses at the output of the D/A is called *impulse sampling* and is not practical. Usually the D/A output is held constant for the full duration between samples. This is called *natural sampling*. The hold function is a form of interpolation or reconstruction filtering that causes an error called *sin x/x error* or *aperture loss*, which is a small (<4 dB) roll-off of the higher frequencies. This error is easily corrected with a small amount of high-frequency peaking called *sin x/x compensation*. This peaking is usually built into the reconstruction filter, but it can also be included within the digital processing section.

Glitch energy is another error associated with the practical hardware implementations of the D/A converters. Glitch energy is noise created during the transition between one sample value and the next. This can be caused by timing skew between the various bits in the binary word or simply by crosstalk from the D/A digital input to the analog output. Glitch energy can be minimized by timing the sample and hold to capture each sample value only after the D/A output has settled, but before the next sample arrives. Isolation between the digital and analog sections of the system is also needed. (See Figures 6 and 7.)

Digital processing

In the simple system, the A/D output was connected directly to the D/A input. Some digital communications or recording systems may be similar to this, but mathematically processing the signal while it is in digital form is what DSP is all about.

FIR and IIR filters

Most filters can be described as causal linear-time invariant (CLTI). *Causal* indicates that the present output depends on the present and the past but not on the future. *Linear* indicates that the output is the result of addition or multiplication of the input by a constant. The input cannot be multiplied by itself so the transfer function is not a function of the input signal. This also means that superposition applies, and no new frequencies are generated. Perhaps most important, it means that the system can be completely characterized by its impulse response. *Time invariant* means the system does not change as a function of time.

All types of CLTI filtering functions such as lowpass, highpass and bandpass, can be implemented using DSP

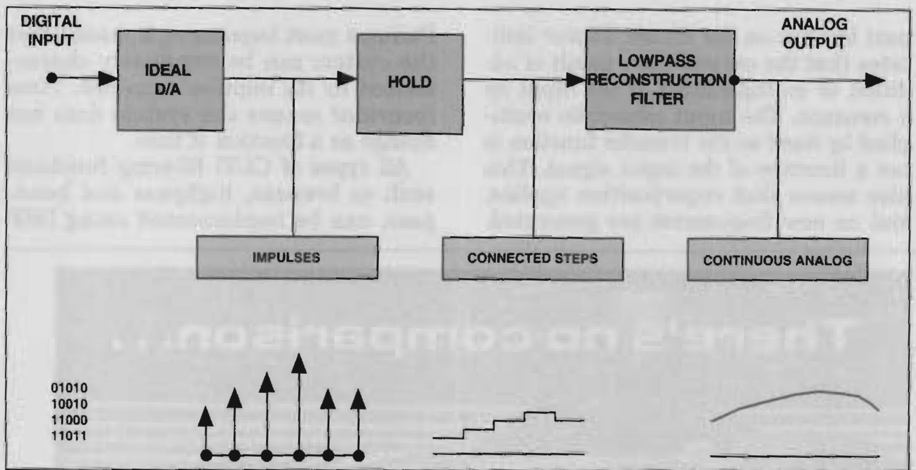

Figure 6. Digital to analog conversion.

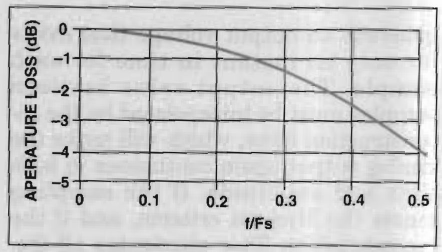

Figure 7.

structures that consist only of delay elements, multiplications by constants and addition. For example, taking the five previous input samples, multiplying each by the constant 1/5 and adding the results creates the average, which is a form of lowpass filter. Various types of filters can be implemented using multiply-accumulate operations that convolve the input signal with the filter's impulse response in the time domain. This is equivalent to multiplication in the frequency domain.

There are two fundamental types of CLTI filters: *recursive* and *non-recursive.* In non-recursive systems, the present output is created using only past and present inputs. (There is no feedback from the output.) A recursive system uses both past and present inputs as well as past and present outputs. (There is feedback.) There are various structures or arrangements for the delays and multiplier-accumulators that produce equivalent results. *Canonic form structures,* which include the direct form, minimize the number of delay elements or registers needed. Other structures, such as the *parallel form* and *cascade form,* require more computation but are less sensitive to coefficient quantizing errors. This is analogous to forms of analog filters that are less sensitive to component tolerances.

The figures show examples of non-recursive and recursive third-order filters with the direct form I structure. The filters contain three delay elements, and the outputs are the summation of four multiplications. The delay elements, represented as Z^{-1} functions, can be implemented as shift registers in hardware or memory locations in software. The signal is advanced once per sample period. The four multiplication constants are represented as C1–C4 where the signal sample value is multiplied by the associated constant.

The series of delay elements can be compared to a tapped delay line. The signals at each delay are often called *taps,* and the multiplication values associated with each tap are often called the *tap weights* or *coefficients*. In a SAW filter, the delays are created by the propagation delay of the acoustical wave along the surface of the substrate, and the tap weights are controlled by the size of the transducer fingers.

When an impulse is applied to the input of a non-recursive filter, the duration of the output or impulse response is equal to the length or number of delay elements within the filter. Thus, the filter has a finite impulse response (FIR). Due to the feedback, recursive filters theoretically can have an infinite impulse response (IIR). The impulse response of practical IIR filters, although longer than the actual length of the filter, does not continue indefinitely but decays exponentially to a negligible value.

Sharp, high-order filtering functions with narrow transition bands require many poles and zeroes corresponding to many Ls and Cs in analog filters. Similarly, sharp filtering in DSP requires many taps. In general, recursive IIR filters require fewer taps to produce a given response. Non-recursive FIR filters, whose impulse response is symmetric in time, have exact linear phase response and therefore constant group delay within the passband, which is important in many applications. An FIR filter can be used to create a Hilbert transform that provides a constant 90° phase shift over a range of frequencies.

Non-recursive FIR filters are unconditionally stable. Recursive IIR filters, however, can be unstable. Recursive filters that are normally stable can be-

NON-RECURSIVE FILTERS

- No feedback from the output
- Unconditionally stable
- Finite impulse response (FIR)
- Impulse response duration equal to filter length
- Response is all zeroes
- Capable of linear phase response without phase equalizer
- Requires more taps for given a response

RECURSIVE FILTERS

- Feedback from output
- Can become unstable
- Usually infinite impulse response (IIR) in theory
- Impulse response duration exceeds filter length in practice
- Response contains poles and zeroes
- Requires fewer taps for a given response

Table 2. Comparison of non-recursive and recursive filters.

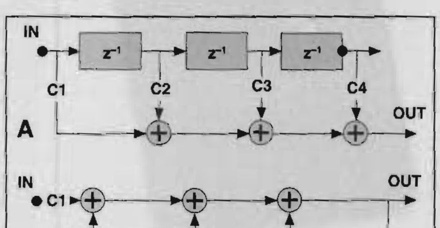

Figure 8. a) Third-order non-recuresive FIR filter. b) Third-order recursive IIR filter.

TYPE OF WINDOW	TRANSITION BANDWIDTH RELATIVE TO RECTANGULAR	FIRST SIDELOBE LEVEL OF WINDOW
Rectangular	1x	−13 db
Bartlett	2x	−27 db
Hanning	2x	−32 db
Hamming	2x	−43 db
Blackman	3x	−58 db

Table 3. Main lobe transition width vs. sidelobes of various windows.

come unstable due to non-linearities caused by momentary overflows or because of roundoff errors recirculating indefinitely. These forms of instability are called *limit-cycle oscillations* and *dead band*. IIR filters can also suffer from other non-linear phenomena such as *jump* and *subharmonic responses*. Jump occurs when the output level changes suddenly due to a small change in the input level. Subharmonic responses occur where the output signal contains subharmonics of the input. These phenomena are similar to the non-linear behavior sometimes exhibited by class C RF amplifiers and other non-linear systems with feedback. These non-linearities can be minimized by error spectral shaping that feeds the quantizing errors back through another set of taps. (See Table 2 and Figure 8.)

Design of FIR and IIR filters

One method of designing an FIR filter to have a given frequency response is to find the corresponding impulse response using Fourier analysis. The coefficients of the direct form FIR filter are simply equal to the impulse response. Unfortunately, the Fourier series of a finite frequency response is infinitely long, and a filter with an infinite number of taps is obviously not practical. Truncating the series causes a ripple in the passband and sidelobes in the stopband caused by the Gibbs phenomenon. Windowing uses a gradual tapered reduction to shorten the series. This reduces the ripples but causes a widening of the transition band. Various windowing functions tradeoff widening of the main lobe and sidelobe attenuation. This is analogous to the tradeoff that must be made in antenna design. Table 3 lists the main lobe transition width vs. sidelobes of various windows. The response of a windowed filter will be the combination (convolution) of the window response and the filter response.

Equiripple is another approach to the design of FIR filters where the maximum passband and stopband ripples are specified, just as in the design of Chebyshev analog filters. An iterative computer algorithm developed by Parks and McClellan can be used to find a set of coefficients that best meets the ripple criteria.

IIR digital filters can be designed by converting an analog design to a digital design. This involves mapping the poles and zeroes of the analog filter from the s domain to the z domain. The matched z transform and impulse invariance method use a mapping that results in distortion of the frequency response, especially near the folding frequency. The bilinear transform uses the arctangent function to create a highly non-linear mapping that eliminates aliasing effects but suffers from frequency warping. Bessel filters that have been converted to IIR filters digital filters by these transforms, no longer retain their linear phase characteristics. Rather than convert an analog filter design to digital, IIR filters can be designed directly using iterative computer algorithms to find the optimum set of coefficients that minimize the mean square error in the frequency response or any other desired parameter.

As in any digital system, signal quantization causes quantization noise. In a digital filter, the tap weights must also be quantized. This coefficient quantization does not add more noise to the signal, but rather causes errors in the desired frequency response, usually in the form of increased ripple in the passband, and a reduction of the maximum attenuation in the stopband. These errors can also cause narrowband IIR filters to become unstable. Arithmetic roundoff or truncation of intermediate results adds noise to the signal and can also cause limit cycle oscillations in IIR filters.

Special techniques can be used to reduce the amount of hardware or computational complexity of filter designs. Multiplication of one variable by another variable requires a large amount of hardware, and practical filters may require 20 to 100 multiplications. Fortunately, filters require multiplications of a variable by a constant, and this requires less hardware. Just as in ordinary decimal arithmetic where multiplication by 10N can be achieved by simply shifting the decimal point by n places, multiplication by 2N can be achieved in binary arithmetic by shifting by n places. Multiplication by any constant can be achieved by the combination of shifting and adding. For example, x6 can be achieved by adding the results of x4 and x2. This technique is called *canonic signed digits*. When this technique is used, one of the goals of the filter design is to select a set of coefficients that provides the desired response while requiring the minimum number of terms.

The design of digital filters boils down to selecting an IIR or an FIR design, selecting the best structure and then determining the correct values for the coefficients. FIR filters are usually used when linear phase response is required, and IIR filters are used when linear phase is not required. In practice, Fs, the desired filter response and other parameters are entered into a DSP computer-aided design (CAD) tool that calculates the required set of coefficients. The CAD tool can then display the actual response that will be obtained using those coefficients including the effects of quantizing and other errors. The designer can then add more taps, change the structure or make other changes as needed until an acceptable response is obtained.

DSP-based FIR and IIR filters can be used for data acquisition filtering and for equalizing communications channels. Instead of using fixed taps weights, adaptive filters and adaptive equalizers use algorithms that automatically adjust the tap weights to change the filter response to minimize errors in the signal. These are slowly time variant systems that typically require 100 to 1,000 signal samples to converge. Because the coefficients are not constants, the multipliers must be capable of multiplying two variables rather than one variable by a constant. Echo cancellation used in telephony is an example of adaptive equalization. Adaptation can also be used as a technique to design fixed filters.

Modulation/demodulation and decimation/interpolation

In communications systems, DSP can be used for modulation and demodulation or frequency translation. This requires that the signal be multiplied by a cos and a -sin carrier signal normally requiring the multiplication of a variable by a variable. The hardware complexity required can be reduced by setting the carrier frequency to exactly Fs/4 so that the cos component of the carrier becomes the series 1, 0, –1, 0, 1 etc. And the -sin component becomes 0, –1, 0, 1, 0 etc. Multiplying by only these factors can be achieved with much less hardware and usually at a greater speed, but it requires that the carrier frequency have a fixed relationship to the sampling rate.

This same technique can also be used for upconversion and downconversion. Again the LO frequency must be exactly equal to Fs/4. Once the signal has been downconverted to an IF and filtered to a limited bandwidth, the sampling rate can then be reduced by a process called *decimation*. A lower sampling rate is desirable because lower-speed hardware can be utilized and because fewer taps are required to implement narrow band filtering. Decimation by an integer factor d simply requires selecting every dth sample and eliminating the rest. The signal must be bandlimited before decimation to avoid aliasing.

Conversely, interpolation is used to increase the sampling rate by an integer factor I by inserting I–1 new samples between the original samples. Initially the inserted samples are all set to 0, and the output spectrum contains a periodic repetition of the original spectrum. The resulting signal is digitally lowpass-filtered, which replaces the inserted zero values with interpolated values and removes the undesired spectral repetitions. By combining interpolation, filtering and decimation, the sample rate can by changed by any factor I/D. Because of the difficult filtering required and the loss of signal amplitude, sample rate conversion by large factors is more efficiently implemented by cascading several stages of conversion and filtering.

DSP using multiple sample rates is known as *multirate signal processing*. Standard filter designs are available that have a fixed response relative to the sampling rate. A halfband lowpass filter, for example, has a -6 dB passband as wide as 1/2 the folding frequency or Fs/4. The number of taps determine the sharpness of the transition band.

FFT and spectral analysis

DSP spectral analysis is often performed using the fast Fourier transform (FFT). The FFT is a particularly efficient algorithm for calculating the discrete Fourier transform (DFT), which is the sampled version of the Fourier transform. The FFT transforms a signal from the time domain to the frequency domain (determines the spectral components of a signal). While conventional analog spectrum analyzers use a single swept filter, the FFT produces a bank of bandpass filters whose outputs are all available simultaneously. FFT algorithms derive their efficiency from breaking the DFT down into successively smaller dfts. The most commonly used FFT algorithm was developed by Cooley and Tukey and requires that the input sequence contain $N = 2^I$ samples in time, where I is an integer. The algorithm requires approximately $N\log_2 N$ complex multiplications and generates n complex output samples, which are usually combined into N/2 real frequency points. For example, using Fs = 1 kHz, 1,024 input points separated by 1 ms spanning a total of 1.024 seconds will produce a Fourier spectrum consisting of 512 frequency points or bins spanning from dc to 511 Hz providing 1

Hz resolution. Note, the adherence to the Nyquist criteria in the highest output frequency bin is 1/2 the sampling frequency. Also, for a given sampling rate, an increase in the number of samples spanning a larger time period, results in better frequency resolution at the output. If a particular application does not require the results at all the output frequency bins, other algorithms such as the Goertzel algorithm may be more efficient. Another alternative, the chirp Z transform, does not require that the number of input points N be a power of 2.

Some signal processing functions are more easily implemented in the frequency domain. In these cases, the FFT can be used to convert time domain signals to the frequency domain, the necessary processing is then performed, and the inverse FFT is used to convert the signals back to the time domain.

The input to the FFT is a finite length sequence representing an input signal. The FFT processes this input as if it were a continuous periodic signal by "splicing" the last time sample to the first time sample. If the actual input signal is periodic and the sampling rate is an integer multiple of the period, then the sampling, called *commensurate,* and output are accurate. If the input signal is not periodic or if the sampling rate is not synchronized to the period of the signal, the spectral output will contain errors, called *leakage,* which are similar to the sidelobes produced by truncating FIR filters. The leakage can be thought of as errors caused by the step change occurring at the mismatched splice point. The leakage can be reduced by windowing the input samples to reduce the amplitude at the splice point, but becuase this effectively shortens the time sample, it also degrades the frequency resolution.

The discrete cosine transform calculates only the real part of the DFT. This efficient transform is useful for data and image compression. The Walsh transform and related Hadamard transform convert to the frequency domain and are based on harmonically related rectangular waves as opposed to the DFT, which is based on harmonically related sine and cosine waves.

The *cepstrum* is an interesting application of spectral analysis. Standard spectral analysis looks for periodicities in the time domain. A signal that has been corrupted by an echo will have peaks and valleys or periodicities in the frequency domain because of constructive and destructive interference. A cepstrum can be thought of as spectral analysis of a spectrum which identifies these periodicities in the frequency domain and relates them back to the time domain. A signal combined with a 1 ms echo will exhibit periodicity at 1 kHz in the frequency domain. The cepstrum will display this frequency domain periodicity as a peak at 1 ms. *Homomorphic deconvolution* is a technique used to identify, separate and sometimes compensate for the effects of a cascaded system. This has been used, for example, to remove the resonance effects of the acoustical horn used to make recordings of Enrico Caruso or to map oil fields using reverberation data.

Many communications signals exist in one dimension, i.e., they vary as a function of time. DSP techniques can also be applied to image processing using multidimensional signals that vary as a function of X and Y location as well as time. **RF**

References

1. Proakis, J. and D. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications*, Macmillan Publishing Company, 1992 New York, NY.

2. Oppenheim, A.V. and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, 1975 Englewood Cliffs, NJ.

3. Ifeachor, E.E. and B.W.Jervis, *Digital Signal Processing, A Practical Approach,* 1993 Addison-Wesley

4. "DSP Digital Signal Processing Databook," 1994 Harris Semiconductor.

About the author

Mark Kolber is a systems engineer with General Instrument (GI) in Hatboro, PA. Prior to joining GI, he worked as an RF engineer for Honeywell in Phoenix, AZ, Aircraft Radio in Boonton, NJ and Blonder Tongue Labs in Old Bridge, NJ. He has a B.S.E.E. from New Jersey Institute of Technology and an M.S.E.E. from Arizona State University. He can be reached at 215-773-8510 or by e-mail at *mkolber@nlvl.com.*