J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

PULSE-SHAPING CIRCUITS

Digital communications systems such as wireless, optical, and wireline telecommunication networks have numerous well-known advantages over analog systems. These systems format binary data into discrete time symbols, modulate the carrier using some form of digital modulation scheme (modulation in this simplified context includes symbol modulation and up-conversion to a carrier frequency) to convert them to transmittable continuous-time signals, and transmit them over a channel. At the receiver the received signal is demodulated (demodulation includes both down-conversion and symbol demodulation), symbol timing is recovered, and the received symbols are reformatted into a usable form. This process is shown in simplified form in Fig. 1.

The discrete time symbols of a digital communication system are of duration T and are sequential, that is, they are orthogonal in time. To preserve the fixed time duration of symbols generated in the transmitter would require infinite bandwidth in the signal processing of the communication system and the communication channel. Finite bandwidth processing causes the symbols to spread in time. This causes the symbols to overlap in time and induces intersymbol interference (*ISI*).

As will be described in the next section, even when finite bandwidth signal processing is used in the communication system, the effects of *ISI* could be eliminated if ideal signal processing components (brick wall filters as an example) could be implemented. This is of course not possible. Pulse shaping is used in the implementation of digital communication systems to minimize *ISI* caused by nonideal channel characteristics and nonideal component implementations.

Pulse shaping is performed in the transmit and receive filters of a communication system. The most common class of filters used for pulse shaping are Nyquist filters (1,2,3). This article gives an overview of the theoretical foundations of pulse-shaping and describes some practical implementations of pulse shaping filters. The material in this article assumes that the communication channel can be modeled as a linear time-invariant filter with additive white Gaussian noise (*AWGN*). For more detailed and in-depth coverage of pulse shaping the reader is referred to Refs. 1 to 3.

Theoretical Background

Ideal data symbols are of fixed duration T. The spectrum for such a symbol is of the form

$$H_s(j\omega) = \frac{\sin[T\,(\omega/2)]}{T\,(\omega/2)} \qquad (1)$$

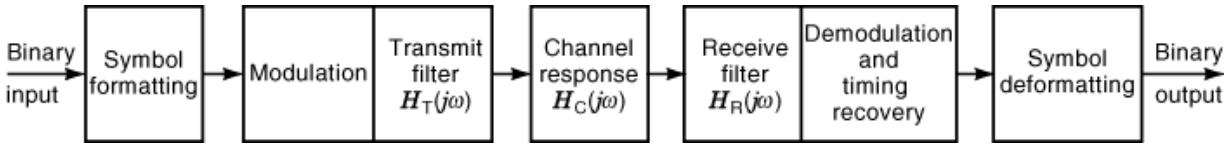

Fig. 1. Simplified block diagram of a digital communication system.

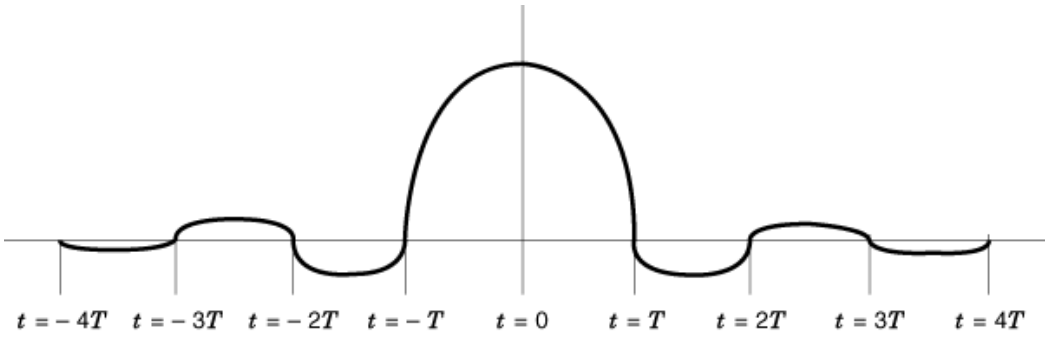

Fig. 2. General form of a pulse shape for an ideally band-limited symbol.

Thus such a symbol would require infinite bandwidth, which is of course impractical. Real communication channels are of fixed bandwidth W. The ideal band-limited channel response (1) is

$$H_C(j\omega) = \begin{cases} 1, & |\omega| < W \\ 0, & |\omega| \geq W \end{cases} \tag{2}$$

The transmit filter, which is designed to shape the transmit symbols to meet the power spectrum constraints of the channel, has an ideal frequency-domain response (1) of

$$H_T(j\omega) = \begin{cases} \pi/\mathrm{W}, & |\omega| < W \\ 0, & |\omega| \geq W \end{cases} \tag{3}$$

The time-domain transmit filter response is therefore

$$h_T(t) = \frac{\sin(Wt)}{Wt} \tag{4}$$

This response is referred to as the pulse shape. $h_T(t)$ is the familiar sinc function, which has zero crossings at integer multiples of Wt except at $Wt = 0$ where it has a value of 1. If we assume a symbol sequence S_n and set $T = \pi/W$, then the pulse for each symbol is of the form shown in Fig. 2 for a single pulse and in Fig. 3 for multiple pulses.

From Fig. 3 it is obvious that if symbols are sampled at times $t = nT$, where n is an integer, the effects of *ISI* are mitigated. This is because the contribution to the composite waveform is zero for all pulses except for the pulse corresponding to S_0 because they are all at a zero crossing point.

The problem is that to produce $h_T(t) = \sin(Wt)/Wt$ requires an ideal brick wall filter, which is impossible to implement. One could attempt to approximate the ideal filter but this would not be cost effective and would

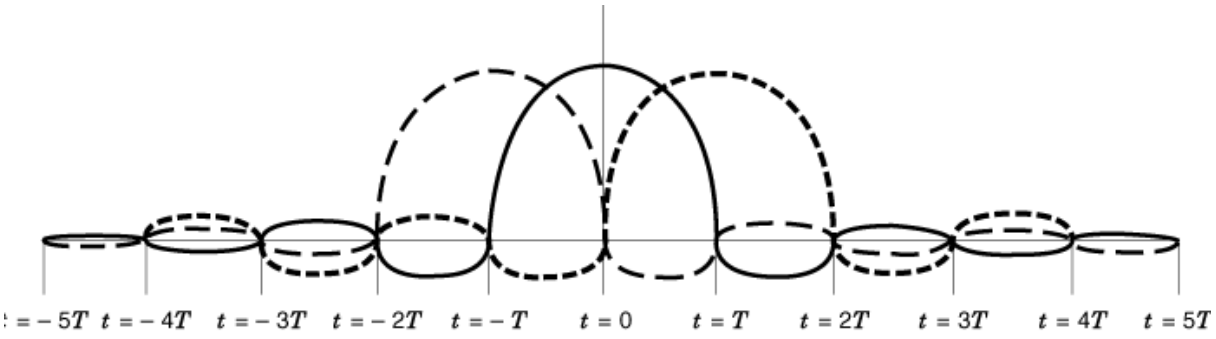

Fig. 3. Multiple pulse shapes for an ideally band-limited symbol.

have other undesirable effects on the system such as making timing recovery difficult (1). To address these problems, more practical pulse shapes have been developed.

The pulse shape at the input to the demodulator, $h_{\mathrm{p}}(t)$, is the convolution of the transmit filter, the channel response, and the receive filter, that is,

$$h_{\mathrm{P}}(t) = h_{\mathrm{T}}(t)^* h_{\mathrm{C}}(t)^* h_{\mathrm{R}}(t) \tag{5}$$

From our preceding discussions it is clear that to eliminate *ISI*, $h_{\mathrm{p}}(t)$ should be forced to cross zero at nonzero integer multiples of T. This can be written as

$$h_{\mathrm{P}}(\mathrm{k}T) = \delta_k \tag{6}$$

The composite input to the demodulator is

$$R(t) = \sum_{\mathrm{n}=-\infty}^{\infty} \mathrm{S_n} h_{\mathrm{P}}(t - nT) + N_{\mathrm{R}}(t) \tag{7}$$

where $N_{\mathrm{R}}(t)$ is the noise input at the demodulator, that is,

$$N_{\mathrm{R}}(t) = N(t)^* h_{\mathrm{R}}(t) \tag{8}$$

From Eq. (1) it can be seen that forcing the composite pulse to cross zero does not necessarily produce an optimal solution because it ignores the noise contribution. Joint optimizations are possible but they are beyond the scope of this discussion. The reader is referred to 1 for in-depth coverage of this subject matter.

Taking the Fourier transform of Eq. (6) and applying the Nyquist sampling theorem, we get

$$\frac{1}{T} \sum_{\mathrm{n}=-\infty}^{\infty} H_{\mathrm{P}}\left(j\omega - n\frac{2\pi}{T}\right) = 1 \tag{9}$$

Equation (9) is referred to as the Nyquist criterion, and filter responses (pulses) that meet this criterion are referred to as Nyquist filters (pulses).

These filters require more than the ideal bandwidth defined in Eq. (3). While simplifying the system implementation, this additional bandwidth requirement costs additional spectrum and noise bandwidth at the

receiver. The additional bandwidth is referred to as excess bandwidth and is usually expressed as a percentage of the ideal bandwidth, that is,

$$B_{\text{excess}} = \left(\frac{B_{\text{actual}}}{B_{\text{ideal}}} - 1\right) \times 100 \tag{10}$$

The actual pulse shape used in a particular application is highly dependent on the targeted channel characteristics. The most commonly used filters satisfying the Nyquist criterion for pulse shaping are raised cosine filters. For this reason raised cosine pulse shaping will be used in this article to describe implementations. Raised cosine filters are of the form

$$h_{\text{P}}(t) = \frac{\sin\left(\frac{\pi \text{t}}{T}\right)}{\frac{\pi t}{T}} \frac{\cos\left(\frac{\alpha\pi \text{t}}{T}\right)}{1 - \left(\frac{2\alpha t}{T}\right)^2} \tag{11}$$

where α controls the rate at which the energy rolls off. The smaller α is, the faster the roll off. Note that for $\alpha = 0$,

$$H_{\text{P}}(j\omega) = \begin{cases} T, & 0 \le |\omega| \le (1-\alpha)\frac{\pi}{T} \\ \frac{T}{2}\left\{1 - \sin\left[\frac{T}{2\alpha}\left(|\omega| - \frac{\pi}{T}\right)\right]\right\}, & (1-\alpha)\frac{\pi}{T} \le |\omega| \le (1+\alpha)\frac{\pi}{T} \\ 0, & |\omega| > (1+\alpha)0\frac{\pi}{T} \end{cases} \tag{12}$$

$$h_{\text{P}}(t) = \frac{\sin\left(\frac{\pi t}{T}\right)}{\frac{\pi t}{T}} \tag{13}$$

which is the ideal pulse shape for a band-limited channel.

In most actual implementations the raised cosine filter is partitioned with part of the response implemented in the transmit filter and part implemented in the receive filter. The most common partitioning is to implement each as the square root of the raised-cosine filter. This is commonly referred to as a root-raised-cosine filter (1) and is given as

$$h_T(t) = h_{\text{R}}(t) = \frac{4\alpha\left(\cos\left((1+\alpha)\frac{\pi t}{T}\right) + \frac{T\sin\left((1-\alpha)\frac{\pi t}{T}\right)}{4\alpha t}\right)}{\pi\sqrt{T}\left[1 - \left(\frac{4\alpha t}{T}\right)^2\right]} \tag{14}$$

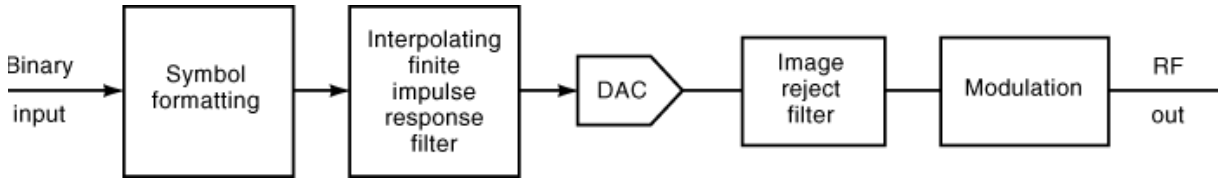

Fig. 4. Pulse-shaping process in a digital transmitter.

and

$$H_T(j\omega) = H_R(j\omega) = \begin{cases} \sqrt{T}, & 0 \le |\omega| \le (1-\alpha)\dfrac{\pi}{T} \\ \sqrt{\dfrac{T}{2}\left\{1 - \sin\left[\dfrac{T}{2\alpha}\left(|\omega| - \dfrac{\pi}{T}\right)\right]\right\}}, & (1-\alpha)\dfrac{\pi}{T} \le |\omega| \le (1+\alpha)\dfrac{\pi}{T} \\ 0, & |\omega| > (1+\alpha)\dfrac{\pi}{T} \end{cases} \tag{15}$$

Implementation

In contemporary digital communication systems it is often not possible to achieve economically the required performance from analog pulse-shaping filters. Shaping filters are therefore implemented using a hybrid approach. The bulk of the shaping is done at baseband using finite impulse response (*FIR*) digital filters, which implement a truncated version of the pulse shape.

On the transmit side of a communication system data symbols are passed to an interpolating digital filter (4). The impulse response of the digital filter typically spans multiple symbol times and is the convolution of the desired pulse-shaping response and precompensation for the roll-off frequency response of the subsequent digital-to-analog converter (*DAC*) if the roll-off is great enough to affect system performance. The output of the digital filter is passed to a *DAC* and a subsequent image reject filter for conversion to continuous-time analog format. The output of the *DAC* in the spectral domain is the baseband spectrum and images of the baseband spectrum occurring within bands, which are bounded by integer multiples of one-half of the sampling rate f_s, all multiplied by a sinc function with nulls occurring at integer multiples of the sampling rate. In the case of low-pass signals the images are present near integer multiples of the sampling rate. In the time-domain the output waveform is "stair stepped." The image reject filter removes all of the spectral images and passes the baseband spectrum, thus "smoothing" or interpolating the time-domain waveform.

At the output of the image reject filter, the shaped symbol pulses are in continuous-time analog format. They can then be up-converted to the desired radio frequency (*RF*) for transmission. This process is shown in Fig. 4.

The operation at the receiver is the transpose of the transmit operation. The analog front end (*AFE*) of the receiver translates the *RF* signal into a convertible pulse train. The pulse train is then converted to a digital form via an analog-to-digital converter (*ADC*). The pulse train is filtered by a decimating *FIR* filter (4) with a root-raised-cosine response to complete the raised-cosine pulse-shaping operation. The receive process is shown in Fig. 5.

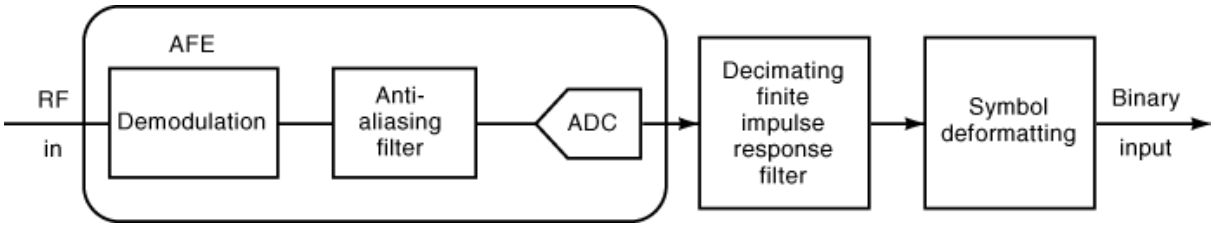

Fig. 5. Pulse-shaping process in a digital receiver.

Digital Pulse-Shaping Network. The actual design procedure for a pulse-shaping network is now presented. A transmit-side root-raised-cosine pulse-shaping network is described. The receive-side network will not be addressed since it is the network transpose of the transmit network.

Assume a digital data transmission system has a symbol period of T seconds, that is, every T seconds a new b_{Q} bit symbol is transmitted. In this description we assume $b_{\mathrm{Q}} = 1$ for simplicity of illustration, although this is not necessary. Before transmission, it is necessary to shape each transmitted pulse. In this case a positive pulse shape is transmitted for a bit value of one, and a negative pulse is transmitted for a bit value of zero. In addition to considering the desired pulse type, it is necessary to determine the length of the pulse relative to the symbol length and the resolution of the pulse or the number of bits used to compute the pulse values.

The interpolating *FIR* filter shown in Fig. 4 must be economically implementable and can therefore only approximate a desired pulse shape, the root-raised cosine given by Eq. (14) in this case. The response is approximated by first determining the desired filter length. The filter length N is the product of the symbol interpolation factor L and the desired span M. The span is the number of symbol periods over which the shaping filter will operate.

The interpolation factor is selected to minimize the $\mathrm{sinc}(x)$ distortion at the upper edge of the band of interest caused by the *DAC* and to simplify the implementation of the image reject filter: the greatest consideration is the latter. As was mentioned previously, $\mathrm{sinc}(x)$ distortion in the band of interest can be compensated for by putting a precompensation function $[1/\mathrm{sinc}(x)]$ in the interpolating filter's response. The higher the interpolation rate, the more distance there is between the band of interest and the first image in the spectral domain. The more distance there is, the simpler the image reject filter. Thus the interpolation factor becomes a trade-off between the complexity of the digital interpolating filter and *DAC* speed, and the complexity of the image reject filter. Typically, an interpolation factor of between 8 and 16 is a reasonable trade.

The span over which the impulse response is approximated depends on the amount of out-of-band energy that can be tolerated. An ideal impulse response begins to approach zero after a number of time-domain side lobes. However, the synthesized impulse is of finite duration. How closely the tails of the response approach zero is dependent on the length of time the impulse is approximated. Out-of-band energy is reduced as the span increases so the trade-off becomes filter order versus out-of-band energy due to time-domain side-lobe truncation. Typical values of the span factor are 4 to 16. A typical root-raised-cosine pulse shaper would have an interpolation factor of 8 coupled with a span of 8 to yield a total filter length of 64.

Filter Coefficient Calculation. There are a number of possible methods for calculating the coefficients of the pulse-shaping filter. Two will be discussed here: the direct and optimal methods.

The Direct Method.. An *FIR* digital filter is of the form

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} a_n z^{-n} \qquad (16)$$

The direct method samples the continuous-time impulse response at appropriate points in time to generate the filter coefficients a. Under the assumption of proper sampling

$$a_n = h(n) \underset{\Leftrightarrow}{Z} H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \cong H_{\mathrm{D}}(z) \qquad (17)$$

where $H_{\mathrm{D}}(z)$ is the desired frequency-domain pulse shape and $N-1$ is the filter order.

As stated previously, the total number of coefficients, N, is related to the span and interpolation factors by $N = LM$. Thus, continuing with the root-raised-cosine example one would simply calculate N evenly spaced values using Eq. (14) to calculate the coefficients a_{n}. Since $h_{\mathrm{T}}(t)$ is an even function about $t = 0$, it is necessary to shift $h_{\mathrm{r}}(t)$ in time such that the sampled version of $h_{\mathrm{T}}(t)$ [denoted by $h(n)$] is evenly symmetric around samples $(N-2)/2$ and $N/2$ to ensure a fully linear phase response (5). Since there are L samples per symbol period T, $h_{\mathrm{r}}(t)$ should be evaluated at integer multiples of T/L. This, along with the time-shift requirement, dictates that the discrete-time values necessary to define $h_{\mathrm{T}}(t)$ are given by

$$t(n) = \left(n + \frac{1}{2} - \frac{N}{2}\right)\frac{T}{L} \qquad (18)$$

Substituting this into Eq. (14) gives the sampled version of $h_{\mathrm{T}}(t)$,

$$h(n) = \frac{4\alpha\left(\cos\left((1+\alpha)\dfrac{\pi t(n)}{T}\right) + \dfrac{T\sin\left((1-\alpha)\dfrac{\pi t(n)}{T}\right)}{4\alpha t}\right)}{\pi\sqrt{T}\left[1 - \left(\dfrac{4\alpha t(n)}{T}\right)^2\right]}$$

$$n = 0, 1, \ldots, N-1 \qquad (19)$$

This process is illustrated in Fig. 6 for a filter of length N equal to 64 and including $4\frac{1}{2}$ side lobes on either side of the main lobe of the root-raised-cosine function. Figure 7 shows the corresponding frequency-domain response of the pulse-shaping filter.

As is the case with any digital filter coefficient design method, once the floating-point coefficients are calculated, they must be converted to fixed-point format if the target digital filter is to be implemented in fixed-point arithmetic. Conversion of coefficients to fixed-point format can be as simple as truncation or rounding or can require sophisticated optimization processes. Let

$$a_n' = h'(n) = Q[h(n)] \qquad (20)$$

be the quantized coefficients. The next step in the coefficient design process is to calculate the frequency response of the quantized filter coefficients and compare it with the desired frequency response to see if the filter is within acceptable error bounds. The frequency response of an *FIR* filter is found by evaluating the filter'sz transform on the unit circle. That is, the magnitude frequency response of the quantized coefficients is

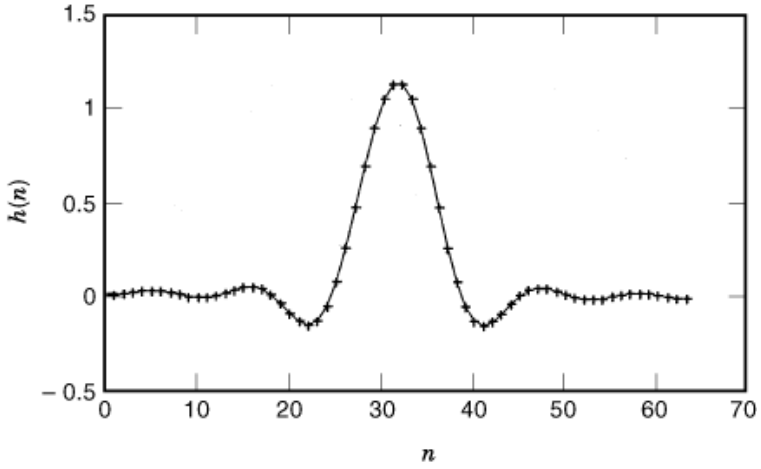

Fig. 6. A discrete time sampled root raised cosine.

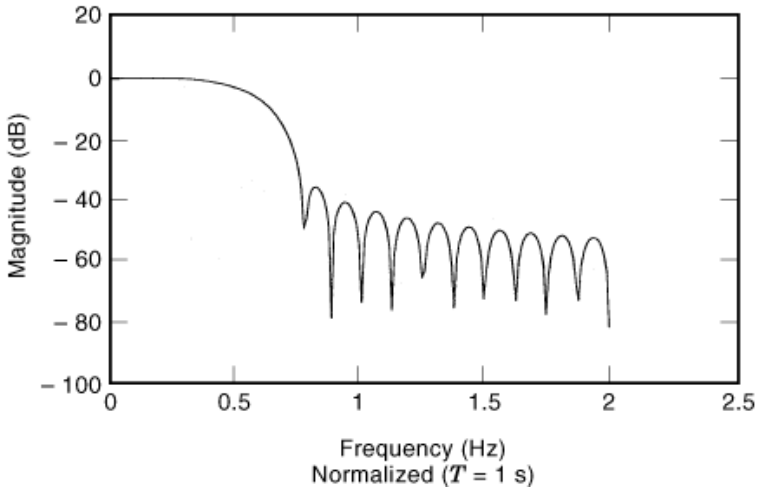

Fig. 7. Frequency-domain response of the sampled root-raised cosine shown in Fig. 6.

given as

$$|H'(z)| = \left| \sum_{n=0}^{N-1} h'(n) z^{-n} \right|_{z=e^{jw}} \tag{21}$$

$H'(z)$ can be evaluated directly or by using a form of the DFT. If the filter's frequency response does not meet the required error bounds, then either the filter order N or the number of bits in the coefficient quantization must be increased and the appropriate steps in the coefficient design process must be repeated.

Coefficient quantization is not as straightforward as data quantization. The quantization of data words yields 6.02 dB of signal-to-noise ratio per bit. The number of bits required for coefficient quantization is a function of filter order and the frequency-response shape. A good rule of thumb for a worst-case metric (6) for

the number of coefficient bits required is

$$b_C = \frac{\text{(Required attenuation in dB)}}{6.02} + \log_2 N \qquad (22)$$

so that

$$\frac{\text{(Required attenuation in dB)}}{6.02} \le b_C \le \frac{\text{(Required attenuation in dB)}}{6.02} + \log_2 N \qquad (23)$$

assuming no limiting roundoff error in the actual filter implementation. In most practical pulse-shaping filter implementations, the coefficient word length should be closer to the lower bound.

The floating-point coefficients (a) for the case $M = L = 8$ are shown in matrix form. Each column represents samples in adjacent symbol periods. Since the impulse response spans eight symbol periods, there are eight corresponding columns.

$$a_n = \begin{bmatrix} -0.010 & -0.002 & 0.042 & -0.061 \\ -0.007 & -0.013 & 0.029 & 0.071 \\ 0 & -0.019 & -0.003 & 0.253 \\ 0.007 & -0.018 & -0.049 & 0.467 \\ 0.014 & -0.010 & -0.101 & 0.689 \\ 0.017 & 0.006 & -0.144 & 0.890 \\ 0.015 & 0.025 & -0.161 & 1.043 \\ 0.008 & 0.039 & -0.137 & 1.126 \end{bmatrix} \begin{bmatrix} 1.126 & -0.137 & 0.039 & 0.008 \\ 1.043 & -0.161 & 0.025 & 0.015 \\ 0.890 & -0.144 & 0.006 & 0.017 \\ 0.689 & -0.101 & -0.010 & 0.014 \\ 0.467 & -0.049 & -0.018 & 0.007 \\ 0.253 & -0.003 & -0.019 & 0 \\ 0.071 & 0.029 & -0.013 & -0.007 \\ -0.061 & 0.042 & -0.002 & -0.010 \end{bmatrix} \qquad (24)$$

In converting the floating-point coefficients to fixed-point coefficients, maximum arithmetic efficiency can be obtained by scaling the fixed-point coefficients so that the maximum possible filter output is at the overflow threshold. Worst-case peak signals would occur for data patterns in which the maxima and minima of the impulse response overlap, or add (in the case of a negative pulse, the absolute value of the minima adds to the composite pulse). It turns out for the example here that an efficient peak value for the impulse response to take is 194 for a 9-bit integer two's complement representation (maximum possible positive value of 255).

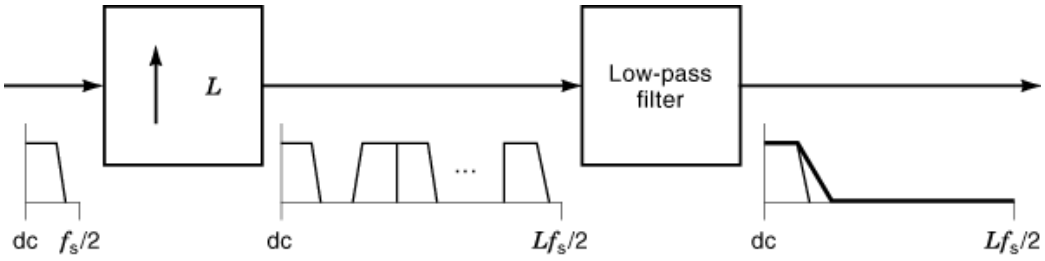

Fig. 8. Illustration of the conceptual interpolation process.

Scaling the matrix of the floating-point values to a peak value of 194 and rounding each coefficient to the nearest integer results in the coefficient set shown here, which completes the synthesis of the pulse coefficients.

$$a'_n = \begin{bmatrix} -2 & 0 & 7 & -10 & 194 & -24 & 7 & 1 \\ -1 & -2 & 5 & 12 & 180 & -28 & 4 & 3 \\ 0 & -3 & 0 & 44 & 153 & -25 & 1 & 3 \\ 1 & -3 & -8 & 80 & 119 & -17 & -2 & 2 \\ 2 & -2 & -17 & 119 & 80 & -8 & -3 & 1 \\ 3 & 1 & -25 & 153 & 44 & 0 & -3 & 0 \\ 3 & 4 & -28 & 180 & 12 & 5 & -2 & -1 \\ 1 & 7 & -24 & 194 & -10 & 7 & 0 & -2 \end{bmatrix} \quad (25)$$

The Optimal Method.. The optimal method requires an optimizing filter design program such as the Parks–McClellan program (5,7) but guarantees an optimum solution. To design the pulse-shaping filter coefficients using the optimal method the EFF subroutine (EFF is the actual name of the subroutine in the published program) in the Parks–McClellan computer design program must be modified by inserting a floating-point, discrete-frequency version of the desired frequency- domain pulse shape. The program is then executed with a desired filter order and the same iterative process of evaluating the frequency response of the quantized coefficient set as described previously for the direct method is followed until a set of coefficients is generated that is within the desired error bound. This method yields much more reliable results than the direct method.

Even more sophisticated filter programs exist that optimize quantized coefficients. These programs allow designers to optimize the efficiency of a filter implementation.

Filter Hardware Implementation. The actual hardware realization of a pulse-shaping filter is highly dependent on symbol rates, filter order, process, etc. Assuming symbol rates that allow implementations in state-of-the-art semiconductor processes, the filter would be implemented as some variation of a polyphase architecture (4).

Interpolation is the process of increasing the sampling rate while preserving the signal's spectral content. The conceptual first step in interpolation is to insert $L-1$ zero-valued samples between each valid input sample, expanding the sampling rate by L. This causes the original signal spectrum to be repeated $L-1$ times. This process is referred to as sample rate expansion. To complete the interpolation, the zero-valued input samples are converted to spectrally accurate approximations of the signal. This is equivalent to preserving the original signal spectrum. Thus, and again conceptually, the zero-stuffed input stream is filtered by a low-pass filter with a passband at the original spectral location and a passband gain of L. This filters out all of the repeated spectra. This conceptual process is shown in Fig. 8.

In real implementations it would be a waste of storage and computational resources to store and multiply zero-valued data samples. For this reason, polyphase structures are used. A polyphase interpolator configures

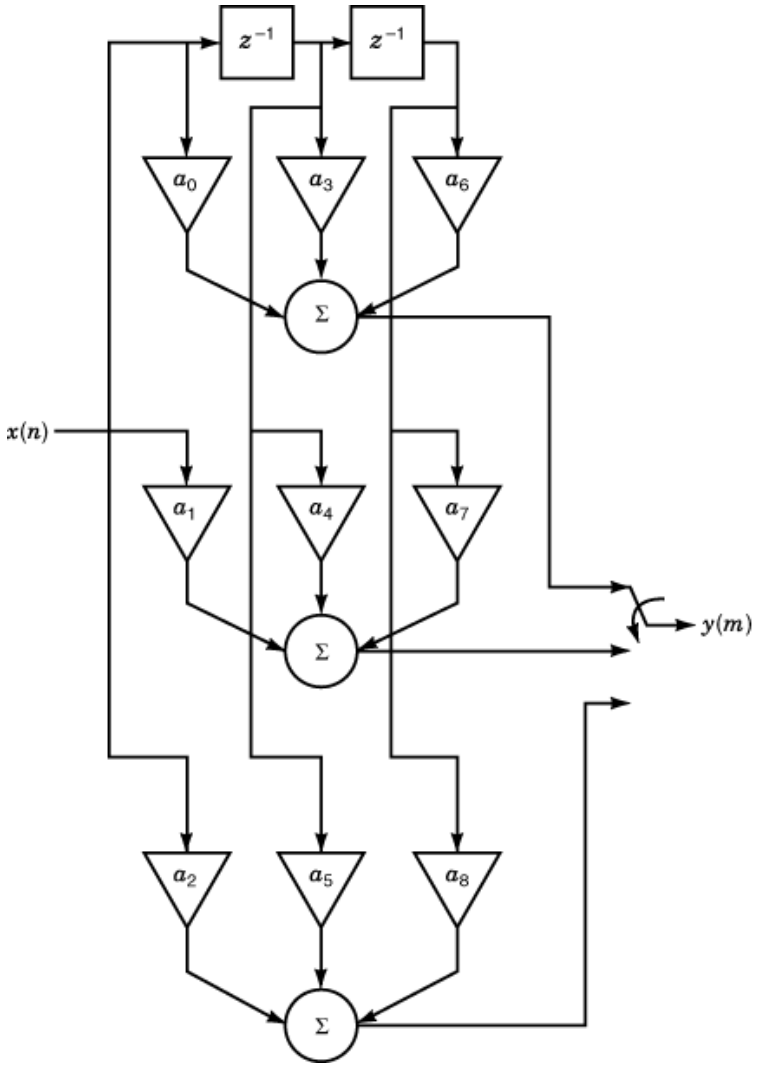

Fig. 9. Polyphase implementation of an *FIR* interpolation filter.

the standard N-tap filter structure into k phases of an $(N-1)$-element delay line. For example, an eighth-order $(N=9)$ *FIR* filter with $k=3$ phases has three sets of three coefficients. The coefficients for the eighth-order filter are $a_0, a_1, \ldots, a_8$. The first polyphase taps the eight element delay line at the input to the filter (a_0), after the third delay (a_3), and after the sixth delay (a_6). The second phase taps a_1, a_4, and a_7. The third phase taps a_2, a_5, and a_8. However, since all but every third element in the delay line would contain a zero-valued data sample, the delay line can be collapsed to a two-element delay line ($N/k-1$ elements) with all phases using the same samples. The output of the interpolator is the commutated output of the three phases. Figure 9 illustrates the polyphase implementation of the nine-tap interpolator. Reference 4 details the efficient implementation of interpolation and decimation filters.

An interpolation filter can be implemented using standard multiply and accumulate elements, or in some cases it is most efficient to use table look up based bit-slice filter methods (8). The trade-off is dependent on

the number of bits required for the representation of a symbol. For low-bit count, the bit-slice implementation is usually more efficient. As bit count increases, table sizes become unmanageable.

DAC

The *DAC* has the task of generating an analog output based on the digital input supplied by the *FIR* filter. The *DAC* must be linear to ensure no unwanted spurious energy is generated. Several implementations are possible depending on the *DAC* resolution required. For relatively modest resolutions of 4 to 8 bits, simple binary weighted current or charge-based approaches are possible. For higher resolutions, it may be desirable to use segmented approaches where each least significant bit (*LSB*) step has a separate current- or charge-based generator. For example, a 5-bit segmented current-steered *DAC* would have 32 separate current sources feeding a common resistor to generate the desired output voltage. Sources are turned on or off depending on the value of the control word. This is opposed to a binary approach in which only five current sources are used scaled in binary fashion. For higher-bit resolutions, the binary approach can suffer from linearity errors due to mismatches at major carries. The advantage of a binary approach is smaller implementation area. For DACs of 9-bit resolution or higher, a combination of binary [for the most significant bits (*MSBs*)] and segmented (for the LSBs) is frequently used.

Standard DACs offer no image suppression, even if the *DAC* runs at a rate greater than the digital word update rate. Image suppression could be improved if the *DAC* could interpolate between successive word updates. Such a converter can be demonstrated, following up on the example begun for the pulse-shaping network. Suppose specifications require that all spurious energy be at least 40 dB down with respect to the baseband level. This requires that all spectral images of the *FIR* filter be attenuated sufficiently to achieve this. Further suppose that the symbol rate is 128 kbaud. The update rate f_{s} of the digital filter is 1.024 MHz (due to the 1:8 interpolation factor from the preceding example), and hence the first image replica will appear there. If the *DAC* were updated at the 1.024 MHz rate, then the amount of attenuation of the image due to roll off is only 17 dB. Consider an interpolating *DAC* operating at five times the digital update rate ($f_{s_1} = 5.12$ MHz). The block diagram of an architecture suitable for monolithic integration accompanied by illustrative spectral plots is shown in Fig. 10. In this architecture, it is shown that interpolation can provide 20 dB of additional attenuation.

A simplified single-ended schematic of an interpolating *DAC* is shown in Fig. 11 (9). ϕ and ϕ_2 are nonoverlapping clocks operating at the interpolation frequency (5.12 MHz in the preceding example). The operational amplifiers are conventional fully differential folded cascade topologies. The input voltage reference is selected based on specific channel requirements and typically ranges from 250 mV to 1 V. It is usually supplied by an on-chip band-gap generator. The configuration is a cascaded charge redistribution *DAC* with the first stage an offset insensitive programmable amplifier to realize the four least significant bits (10). Gain is determined by the sum of the binary weighted capacitors at the input, dependent on the LSB word. The second stage is a modified second-order switched-capacitor biquad (11). This stage has programmable gain to realize the four most significant bits of the *DAC* and acts as the summing point for the LSB portion. A straightforward charge-transfer analysis yields the low- frequency output level as a function of the reference voltage V_{ref} and critical circuit capacitor values

$$V_{\text{out}} = \frac{C_3}{16C_3}\frac{V_{\text{ref}}}{C_1}(b_7C_1 2^{-4} + b_6C_1 2^{-3} + \cdots + b_4C_1 2^{-1}) + \frac{V_{\text{ref}}}{C_7}(b_3C_7 2^{-4} \cdots + b_0 C_7 2^{-1}) \tag{26}$$

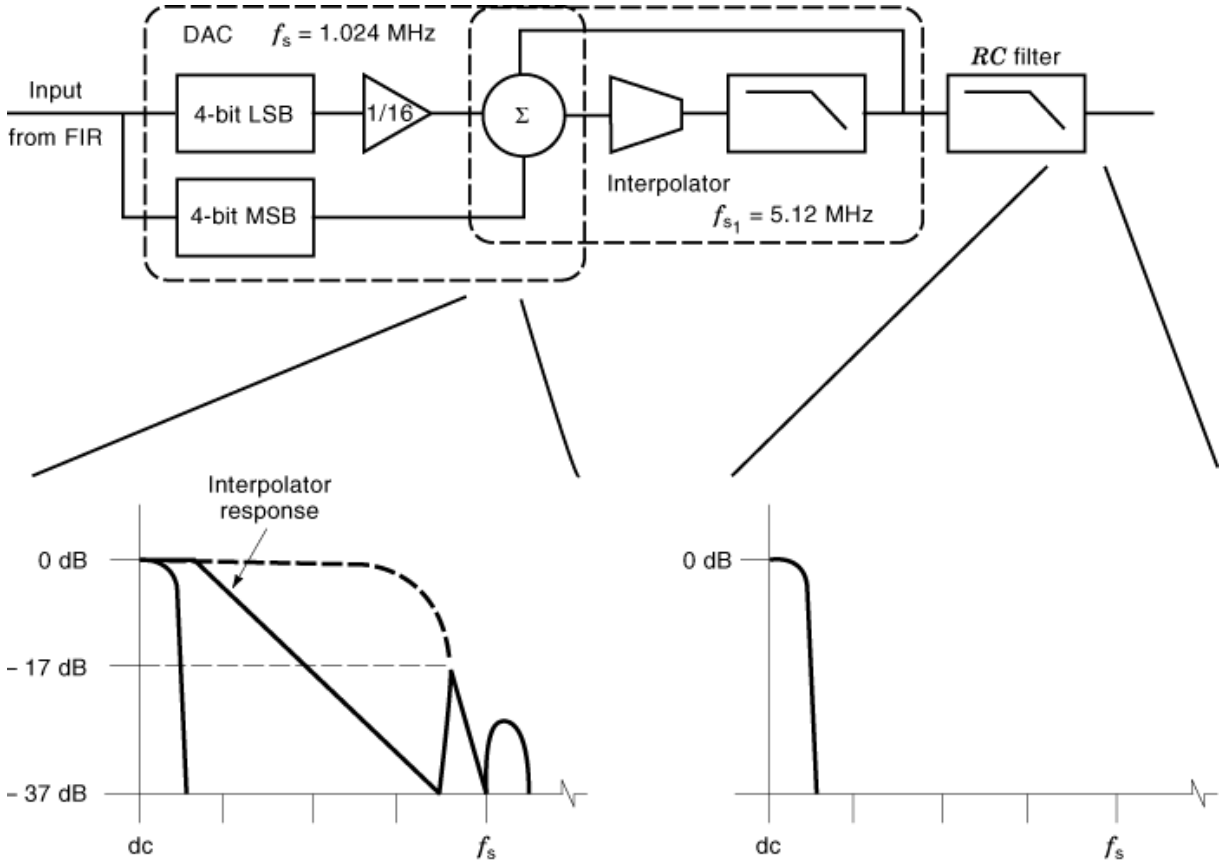

Fig. 10. Digital-to-analog converter top-level block diagram.

Here, $b_0, b_1, \ldots, b_7$ represent the digital data (possible values of 0 or 1) with b_7 as the least significant bit. Since the *DAC* is fully differential, the sign bit is easily incorporated by cross-coupling the input connection to V_{ref}. Interpolation is accomplished by the fact that the final stage is really a switched-capacitor filter. dc and ac analysis is simplified by replacing the *DAC* capacitor array structure with an equivalent single-ended switched-capacitor network as shown in Fig. 12. Including the input offset voltages V_{off_2}, V_{off_3} for the operational amplifiers, the defining charge equations for this network are given by

$$[V_{o_3}(n) - v_{off_3}](1 + \alpha_2) + [V_{o_2}(n) - v_{off_3}]\alpha_6 = V_{o_3}(n-1) - v_{off_3} \tag{27}$$

$$V_{o_2}(n) - v_{off_2} - \alpha_5 v_{off_2} - \alpha_3 v_{off_2} = [V_{in}(n-1) - v_{off_2}]\alpha_5 + [V_{o_3}(n-1) - v_{off_2}]\alpha_3 + V_{o_2}(n-1) - v_{off_2} \tag{28}$$

where V_{o2} and V_{o3} are outputs of amplifiers 2 and 3, respectively.

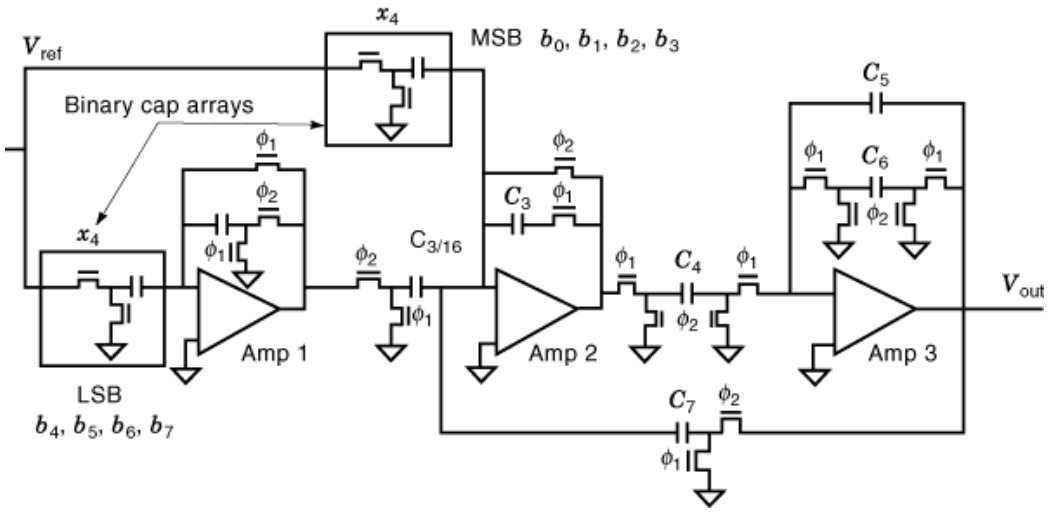

Fig. 11. Simplified single-ended *DAC* schematic.

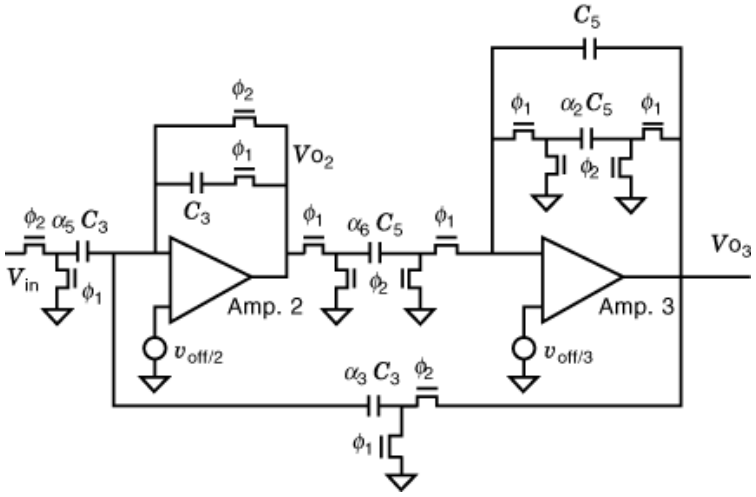

Fig. 12. Interpolating filter.

Under steady-state conditions $n \to \infty$. With $V_{in}(n) = V_{in}(n-1) = 0$, then

$$V_{o_2}(n) \to V_{o_2}(n-1) \to V_{o2}, \qquad V_{o_3}(n) \to V_{o_3}(n-1) \to V_{o3} \tag{29}$$

where V_{o2} and V_{o3} are the dc offset voltage at the outputs of amplifiers 2 and 3, respectively. Under these conditions, solving Eqs. (27) and (28) yield the output offset voltages for the *DAC*,

$$V_{o3} = 0 \tag{30}$$

$$V_{o2} = v_{\text{off}_3}\left(1 + \frac{\alpha_2}{\alpha_6}\right) \tag{31}$$

where Eq. (31) is valid during high ϕ_1. It is interesting to note that although the *DAC* output is a fully held signal over the 5.12 MHz clock period, it has no operational-amplifier-induced dc offset. This is simply due to the fact that output offset at this point is due solely to amplifier 2, which has been nulled out.

Ignoring offset, Eqs. (27) and (28) may be transformed to the z domain, where the following transfer function may be derived:

$$\frac{V_{o_3}(z)}{V_{\text{In}}(z)} = \frac{-\alpha_5\alpha_6}{1+\alpha_2} \frac{z^{-1}}{1 - \dfrac{2+\alpha_2-\alpha_3\alpha_6}{1+\alpha_2}z^{-1} + \dfrac{1}{1+\alpha_2}z^{-2}} \tag{32}$$

This is the form of a second-order low-pass filter. Interpolation is accomplished by designing the filter such that little passband loss occurs at the digital filter's 3 dB point of 64 kHz. Setting specifications for the interpolating filter at no more than 0.3 dB loss at 100 kHz; then with a sampling frequency of 5.12 MHz, this leads to a desired transfer function of

$$H(z) = \frac{0.091\,021\,8z^{-1}}{1 - 1.492\,134\,6z^{-2} + 0.583\,156\,4z^{-2}} \tag{33}$$

Coefficient matching with Eq. (32) yields the necessary capacitor values. With 17 dB of loss due to the S/H effect and the loss at 1.024 MHz due to Eq. (33), the image at the *DAC* output is 37 dB lower. Hence, an additional 20 dB of image rejection is achieved with the use of interpolation techniques.

Image Filter. There are several techniques that are used to realize continuous-time monolithic filters. Standard active filter techniques utilizing resistors and capacitors are the simplest approach but pole frequencies cannot be accurately placed due to typical process variations. If the image response is located very far away from the baseband spectrum, then accurate pole placement may not be necessary. In that case, the filter is simply designed such that the worst-case *RC* product leading to a high cutoff frequency is still sufficient to meet stopband energy requirements. Frequently, only first-order filters are required, but if the passive variation is significant enough, higher-order filters may be necessary to provide sufficient attenuation. If a process has very linear components, such as thin-film resistors and double polysilicon or metal capacitors, it is possible to realize wide dynamic range filters with excellent linearity. Dynamic ranges over 93 dB are possible if linearity is a concern. If only diffused resistors or depletion capacitors are available, then the corresponding voltage dependence of the passive devices reduces linearity considerably. In that case, interpolating *DAC* techniques may be desirable such that only a simple external *RC* filter is necessary for image filtering. Of course linearity requirements must be considered to make a final decision.

Another technique that may be used includes MOSFET-C (metal oxide semiconductor field-effect transistor capacitance) filters (12), where MOSFET transistors operated in the linear region are used to replace resistors in active filters. Using special linearization techniques, MOSFET-C filters can achieve reasonable linearity (50 dB to 60 dB) and no special resistor material is required. However, MOSFET-C filters are limited in the frequency range that they can operate due to operational amplifier limitations and distributed parasitics (13). They also require the use of tuning networks to ensure that the MOSFET devices stay in the linear region and close to the resistor values necessary to achieve the necessary time constants.

Finally, transconductance-*C* (g_m–*C*) networks can be used as image filters (14). g_m–*C* filters have the advantage of being able to achieve very high cutoff frequencies, typically in the 10 MHz to 100 MHz area. These may be useful in cases where data rate is very high and image frequencies may be close to baseband.

Linearization techniques can achieve dynamic ranges in the area of 50 dB to 65 dB. g_m–C filters have pole frequencies proportional to g_m/C, so stable transconductance networks are required to achieve stable pole locations. Usually, this implies some sort of tuning network to accurately set gm for the desired cutoff.

Summary

In this article an overview of the motivation, theory, and implementation of pulse shaping in digital communication systems has been presented. Digital communication systems transmit digital data as discrete-time symbols modulating a continuous-time waveform. Because actual communication channels are band limited, symbol times can overlap, inducing intersymbol interference. Pulse shaping can reduce or eliminate intersymbol interference if it is closely matched to the characteristics of the channel.

A theoretical background for pulse shaping was established based on a channel that can be modeled as a linear time-invariant filter with additive white Gaussian noise. It was shown that for this type of channel a class of filters known as Nyquist filters are the most commonly used for pulse shaping and the most popular filter in this class is the raised cosine filter. Since the implementation of pulse shapers are commonly distributed between the transmit and receive sides of a communication system, an example of the implementation of a transmit-side square-root-raised-cosine filter was presented.

The square-root-raised-cosine filter implementation presented was a hybrid analog and digital implementation, taking advantage of the strengths of both technologies.

BIBLIOGRAPHY

1. E. A. Lee D. G. Messerschmitt, *Digital Communication*, 2nd ed., Norwell, MA: Kluwer Academic, 1994.
2. B. Sklar, *Digital Communications, Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
3. R. E. Ziemer R. L. Peterson *Digital Communications and Spread Spectrum Systems*, New York: Macmillan, 1985.
4. R. E. Crochiere L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
5. L. R. Rabiner B. Gold, *Theory and Application of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
6. G. A. Mian A. P. Nainer, On the performance of optimum linear phase low-pass *FIR* digital filters under impulse response coefficient quantization, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-29**: 925–932, 1981.
7. T. W. Parks McClellan, A program for the design of linear phase finite impulse response digital filters, *IEEE Trans. Audio Electroacoust.*, **AU-20**: 195–199, 1972.
8. F. J. Taylor, *Digital Filter Design Handbook*, New York: Dekker, 1983.
9. B. A. Myers et al., A frequency agile monolithic QPSK modulator with spectral filtering and 75 Ω differential line driver, *IEEE J. Solid State Circuits*, **33**: 1394–1405, 1998.
10. R. Gregorian G. Temes, *Analog MOS Integrated Circuits for Signal Processing*, New York: Wiley, 1986, pp. 414–416.
11. R. Gregorian, Switched capacitor filter design using cascaded sections, *IEEE Trans. Circuits Syst.*, **CAS-27**: 515–521, 1980.
12. Y. Tsividis, M. Banu, J. Khoury, Continuous-time MOSFET-C filters in VLSI, *IEEE J. Solid State Circuits*, **SC21**: 15–30, 1986.
13. J. Khoury Y. Tsividis, Analysis and compensation of high frequency effects in integrated MOSFET-C continuous-time filters, *IEEE Trans. Circuits Syst.*, **CAS-34**: 862–875, 1987.
14. K. Laker W. Sansen, *Design of Analog Integrated Circuits and Systems*, New York: McGraw-Hill, 1994, pp. 652–657.

BRENT A. MYERS
DAVID B. CHESTER
Intersil Corporation