Equalization Concepts: A Tutorial

Application Report

David Smalley
Atlanta Regional Technology Center

SPRA140
October 1994

PRINTED WITH SOY INK ™

TEXAS INSTRUMENTS

Printed on Recycled Paper

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Introduction

DSP-based equalizer systems have become ubiquitous in many diverse applications including voice, data, and video communications via various transmission media. Typical applications range from acoustic echo cancelers for full-duplex speakerphones to video deghosting systems for terrestrial television broadcasts to signal conditioners for wireline modems and wireless telephony. The effect of an equalization system is to compensate for transmission-channel impairments such as frequency-dependent phase and amplitude distortion. Besides correcting for channel frequency-response anomalies, the equalizer can cancel the effects of multipath signal components, which can manifest themselves in the form of voice echoes, video ghosts or Rayleigh fading conditions in mobile communications channels. Equalizers specifically designed for multipath correction are often termed *echo-cancelers* or *deghosters*. They may require significantly longer filter spans than simple spectral equalizers, but the principles of operation are essentially the same.

The literature is rich with practical and theoretical treatments of the various equalization schemes. This article attempts to familiarize you with some basic concepts associated with channel equalization and data communication in general. It is hoped that the liberal use of signal plots will lead to an intuitive understanding of such concepts as intersymbol interference and multipath effects. To this end, the Mathcad 4.0 [15] files used to create the figures have been made available. See the *Code Availability* section on page 174. You are encouraged to experiment further with these files. For a more rigorous mathematical treatment, refer to the numerous books and articles cited on page 174. Of particular note is the excellent tutorial by Shahid Qureshi [1], after which this article is loosely patterned.

Of particular interest today is the area of digital cellular communications, which has seen wide use of fixed-point DSPs such as the TMS320C5x. This family of processors provides the processing power to perform the requisite adaptive equalization while at the same time handling such tasks as channel coding, error correction (Viterbi algorithm), and vocoding functions (VSELP), thus providing a highly integrated and yet flexible solution to baseband processing. The last section of this paper provides a brief survey of adaptive equalization for digital cellular systems. For a detailed application example, please see the application report *Channel Equalization for the IS-54 Digital Cellular System With the TMS320C5x* on page 177.

What Is Intersymbol Interference?

Consider what happens when pulsed information is transmitted over an analog channel such as a phone line or airwaves. Even though the original signal is a discrete time sequence (or a reasonable approximation), the received signal is a continuous time signal. Heuristically, one can consider that the channel acts as an analog low-pass filter, thereby spreading or smearing the shape of the impulse train into a continuous signal whose peaks relate to the amplitudes of the original pulses. Mathematically, the operation can be described as a convolution of the pulse sequence by a continuous time channel response.

The operation starts with the convolution integral:

$$r(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \qquad \textbf{(1)}$$

where r(t) is the received signal, h(t) is the channel impulse response, and x(t) is the input signal. The second half of the equation above is a result of the fact that convolution is a commutative operation.

Component x(t) is the input pulse train, which consists of periodically transmitted impulses of varying amplitudes. Therefore,

$$x(t) = 0 \ for \ t \neq kT \tag{2}$$

$$x(t) = X_k \ for \ t = kT \tag{3}$$

where T represents the *symbol* period. This means that the only significant values of the variable of integration in the above integral are those for which τ = kT. Any other value of τ amounts to multiplication by 0. Therefore r(t) can be written as

$$r(t) = \sum_{k=-\infty}^{\infty} x_k h(t - kT) \tag{4}$$

This representation of r(t) more closely resembles the convolution sum familiar to DSP engineers. Note, however, that it still describes a continuous time system. It shows that the received signal consists of the sum of many scaled and shifted continuous time system impulse responses. The impulse responses are scaled by the amplitudes of the transmitted pulses of x(t).

As an example, consider the calculation for r(t) at some noninteger time index (t = 1.1) :

$$r(1.1) = \cdots + x_{-2}h(1.1 + 2T) + x_{-1}h(1.1 + T) + x_0 h(1.1) + x_1 h(1.1 - T) + x_2 h(1.1 - 2T) \cdots \tag{5}$$

One can see how received values for any time t are computed. Each pulse value of the input sequence, x_k, contributes a component of the output summation.

Because you are interested in processing the received signal on digital hardware, you must represent the received signal as a difference equation. Physically, you are periodically sampling the received waveform. For the case of pulse-amplitude modulation, it is sufficient to sample the received signal at the symbol transmit rate, 1/T[2]. (In some instances it can be advantageous to sample at a multiple of the symbol rate to implement a *fractionally spaced* signal processing system.) To represent the sampling mathematically, replace t with nT, where, again, T is the symbol transmit rate:

$$r(nT) = \sum_{k=-\infty}^{\infty} x_k h(nT - kT) \tag{6}$$

which can also be written as

$$r(nT) = x_n h(0) + \sum_{k \neq \infty} x_k h(nT - kT) \tag{7}$$

One last factor to account for is sampling phase. Unless the sample clock is perfectly synchronized with the transmit clock, the sample-phase offset will be nonzero. To account for an arbitrary phase offset in the equation above, add an offset t_0 to the time index.

$$r(nT + t_0) = x_n h(t_0) + \sum_{k \neq \infty}^{N} x_k h(t_0 + nT - kT) \tag{8}$$

In the equation above, the first term is the component of r(t) due to the Nth symbol. It is multiplied by the center tap of the channel-impulse response. The other product terms in the summation are intersymbol interference (ISI) terms. The input pulses in the neighborhood of the Nth symbol are scaled by the appropriate samples in the tails of the channel-impulse response. Below are numerical examples for various values of n with t_0 = 0.1 for values of k spanning the five sample neighborhoods around n.

$$r(0.1) = x_0 h(0.1) + x_{-2} h(2.1) + x_{-1} h(1.1) + x_1 h(-0.9) + x_2 h(-1.9) \cdots (n = 0) \tag{9}$$

$$r(1.1) = x_1 h(0.1) + x_{-1} h(2.1) + x_0 h(1.1) + x_2 h(-0.9) + x_3 h(-1.9) \cdots (n = 1) \tag{10}$$

$$r(2.1) = x_2 h(0.1) + x_0 h(2.1) + x_1 h(1.1) + x_3 h(-0.9) + x_4 h(-1.9) \cdots (n = 2) \tag{11}$$

Figure 1 illustrates a pulse train to be transmitted. The center pulse is x_0, the pulse at 1 is x_1, the pulse at –1 is x_{-1}, etc. If you assume an arbitrary impulse response for the transmission channel, you can construct the received signal r(t). This signal is shown superimposed on the transmit waveform x(t). In actuality, the received waveform would be time shifted because of the channel delay, but for clarity r(t) is shown with no delay relative to x(t). Note that the peaks of r(t) roughly relate to the sense of the corresponding transmit pulses; however, the value of r(t) at the sample instants can be quite different from those transmitted. This is because of ISI effects.

Figure 1. A Pulse Train to Be Transmitted

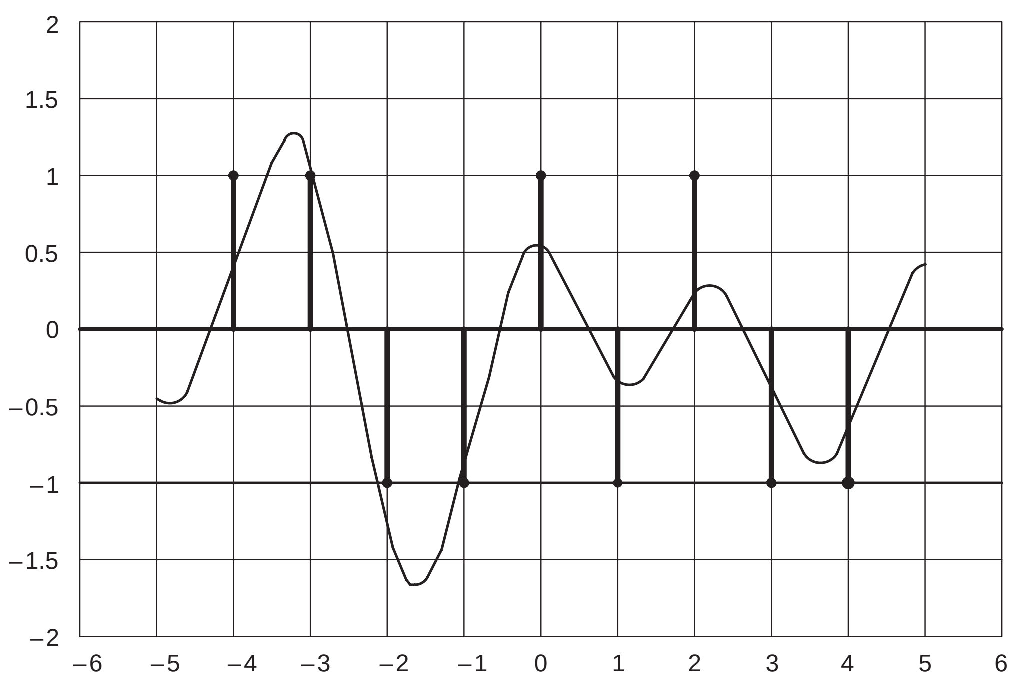
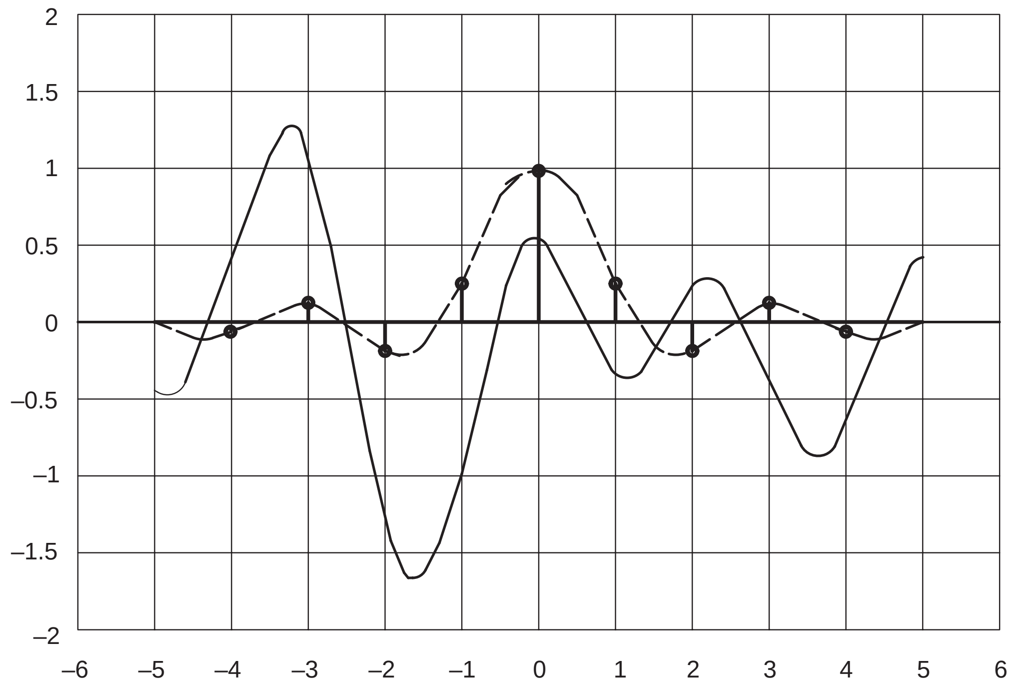


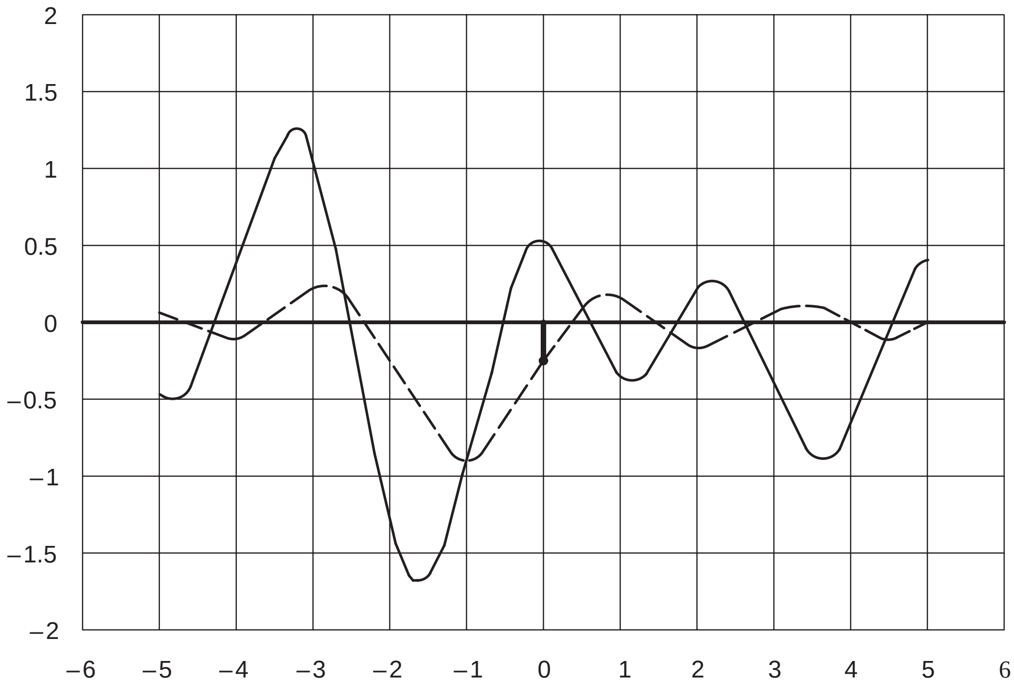
Figure 2 shows the component of r(t) due to a single input pulse x_1, which is superimposed on the received signal r(t). Recall that the shape of this component is the same as that of the transmit-channel impulse response. The values of this individual pulse response at the sample periods (which are multiples of T) are indicated by the black dots. Note that although the signal component in this example is sinc shaped, the nulls *do not* occur at the sample interval. Therefore, the pulse response centered at t=0 makes undesirable contributions to the neighboring received samples of r(t). The contribution of the x_0 symbol to r(0) is the value + 1.

Figure 2. Component of r(t)



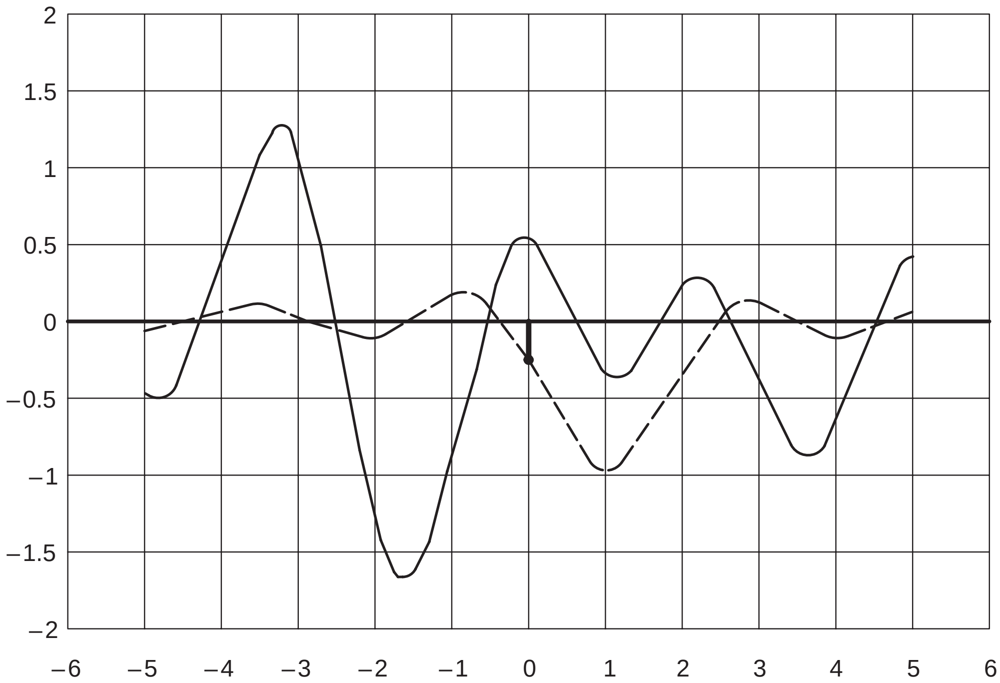
To determine the value of the received signal at t=0, r(0), sum the contributions of the received impulse responses due to x_0, x_{-1}, x_1, x_{-2}, x_2... .

Figure 3. Contribution Due to x_{-1}



As shown in Figure 3, the contribution due to x_{-1} is the value at t=0 of the scaled and shifted impulse response corresponding to the x_{-1} transmit pulse. In this case the impulse response is scaled by –1, which is the value of x_{-1} and is advanced by one sample period because x_{-1} is transmitted one period prior to x_0. Therefore, the x_{-1} symbol results in a small negative component of r(0).

Figure 4. Contribution Due to x_1 at t=0



Similar reasoning explains the contribution due to x_1 , except this time use the value of the time-delayed impulse response at t=0 as illustrated in Figure 4.

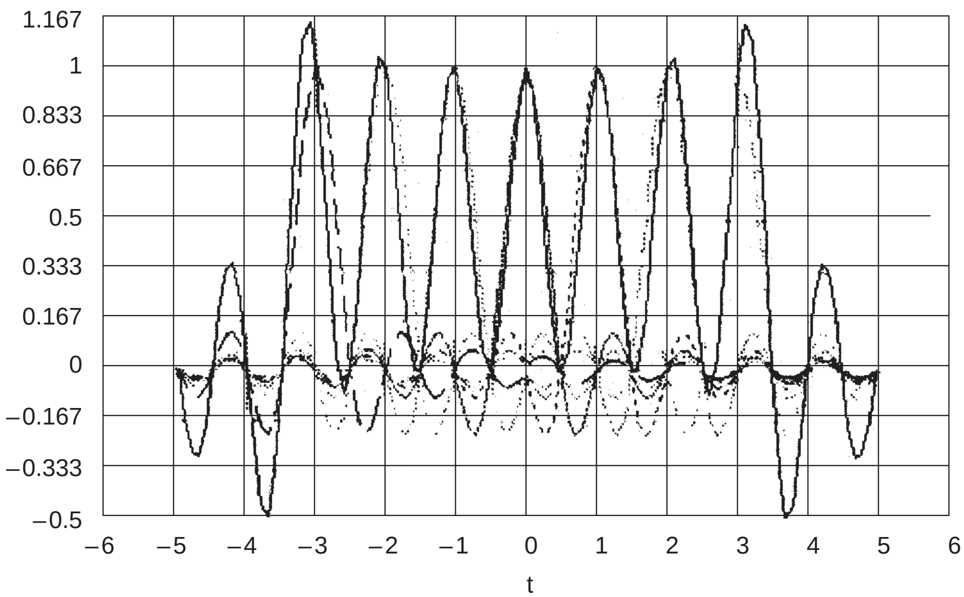
The received value of r(t=0) is computed by summing the contribution of x_0 plus all of the ISI terms; that is, $x_{+/-1}$, $x_{+/-2}$, $x_{+/-3}$, ... Theoretically, this is an infinite sum, but as shown here, the channel response is typically a decaying exponential. Therefore, in practice, an FIR system can be used to model and compensate the system.

From the example above you can see that the nth received sample is primarily influenced by the nth symbol transmitted; however, there are ISI components contributed by prior and subsequent transmit symbols. The terms due to prior symbols (x_{n-1} and before) are termed *postcursor* ISI [3] because the nth transmitted symbol affects on symbols following the nth received symbol. The nature of this ISI can be determined by examining the right-hand portion of the system impulse response. Alternately, the ISI terms due to subsequent transmit symbols (x_{n+1} and beyond) exert *precursor* ISI [3] because the nth transmit symbol influences received symbols prior to the nth. These ISI terms are determined by the shape of the left-hand portion of the system impulse response.

Pulse Shaping

From the preceding figures, it is apparent that ISI is caused when the tails of the received pulses overlap at the sample points, causing uncertainty in the received pulse amplitude. It is possible to shape the transmit pulses in a manner designed to minimize the effects of ISI on the received waveform. As shown in Figure 5, the set of shifted pulse responses overlap, but their tails all possess nulls at the sample instants. Therefore, the only contribution to r(nT) is due to the nth transmit pulse. As shown below, the received signal r(t) equals the amplitude of the individual sinc functions at the sample instants. Compare this with the previous example in which r(t) has a more ambiguous relationship to the individual pulse responses.

Figure 5. Set of Shifted Pulse Responses



If a received pulse shape can meet the following property, zero ISI can be achieved:

$$p_r(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{12}$$

Equation (12) simply means that there are zero crossings at the sample rate. It can be shown that this results in a spectrum possessing *vestigial symmetry.* That is, the frequency response exhibits odd symmetry about 1/2T, causing the sum of repeat spectra to equal a constant. It is important to note that this spectrum may be closely approximated by a realizable filter having a gradual rolloff around 1/2T.

Figure 6. Odd Symmetry

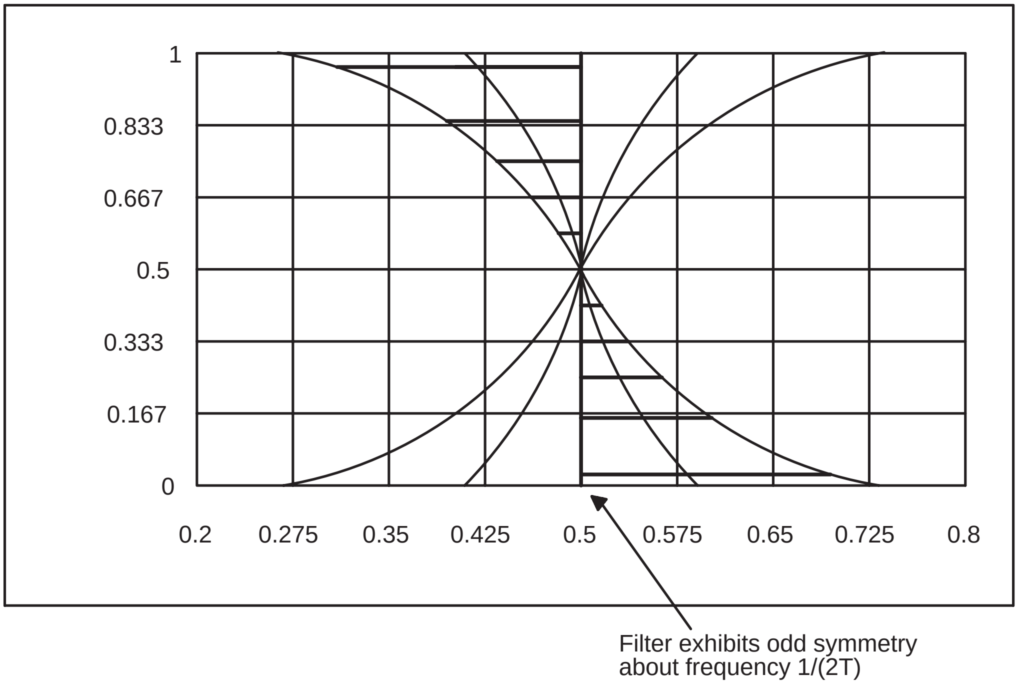


Figure 6 illustrates the notion of odd symmetry.

Figure 7. Spectral Response at 1/(2T)

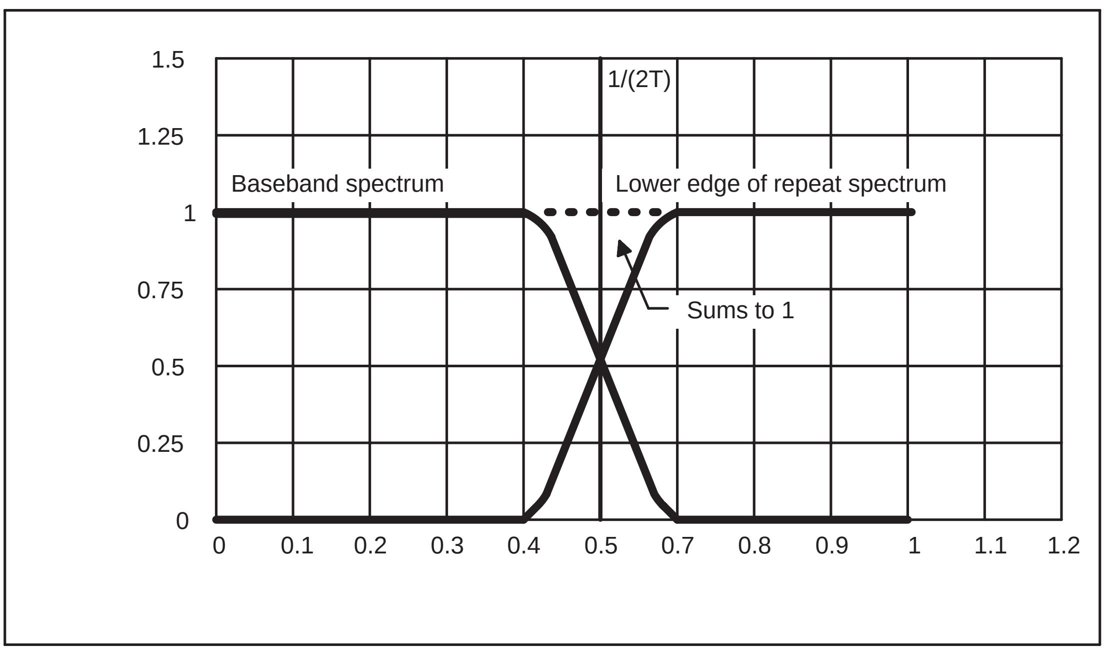


Figure 7 shows the spectral response around 1/(2T). The repeat spectra centered at 1/T actually overlaps the baseband spectrum, but as long as the sum of the two responses is constant, the criterion for zero ISI is met.

One class of linear phase filters possessing vestigial symmetry is the *raised cosine* family:

$$P_{rc}(f) = \begin{cases} T & |f| \le 1/(2T) - \beta \\ \frac{1}{2}\, T\left[1 + cos\left(\frac{\pi|f| - 1/(2T) + \beta}{2\beta}\right)\right] & 1/(2T) - \beta < |f| \le 1/(2T) + \beta \\ 0 & |f| > 1/(2T) + \beta \end{cases} \tag{13}$$

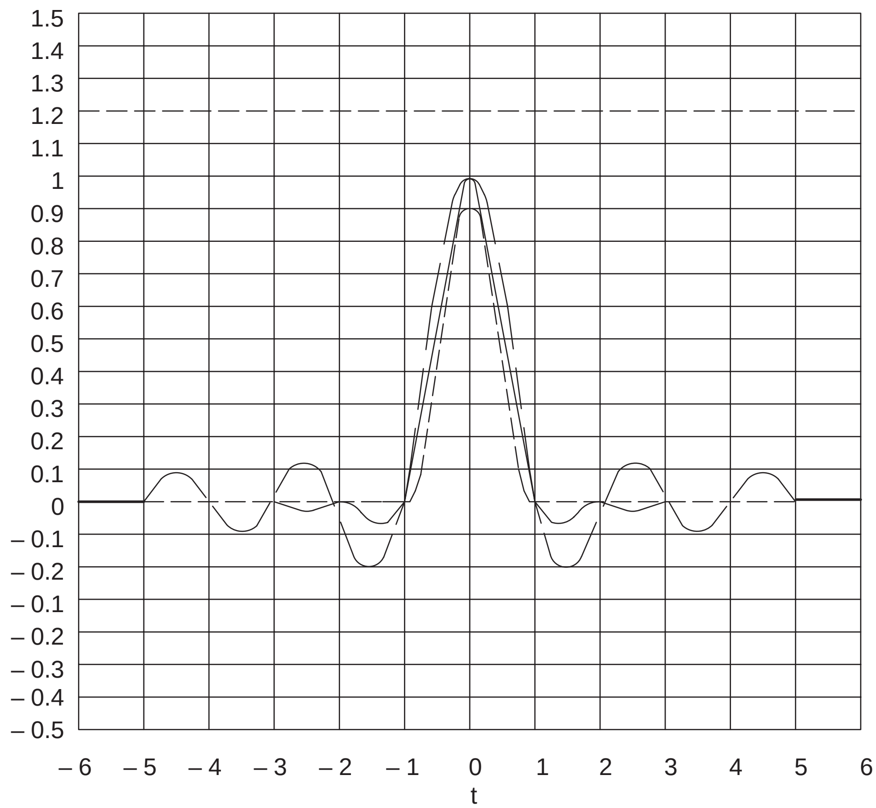
This filter is flat up to $1/(2T)-\beta$ and 0 beyond $1/(2T)+\beta$. The complicated part of the equation above describes the shape of the odd symmetric transition band. Closer inspection of the equation for the transition band quickly reveals the shape of the signal. It is really the cosine of an argument ranging from 0 to π with a DC offset of +1, hence *raised cosine*. The other variables scale the backwards S shape in the x and y dimension to fit the curve into the flat portions of the response.

The impulse response of the signal possessing the raised cosine spectrum is as follows:

$$P_{rc}(t) = \frac{cos 2\pi\beta t}{1-(4\beta t)^2}\, sinc\left(\frac{t}{T}\right) \tag{14}$$

Note that the Equation above can be broken into two parts: the familiar sinc function, which insures that the product will have nulls at multiples of T, and a second term that is an exponentially decaying sinusoid whose rate of decay is proportional to β. The time response of the raised cosine signal for various values of β is shown in Figure 8.

Figure 8. Time Response of the Raised Cosine Signal



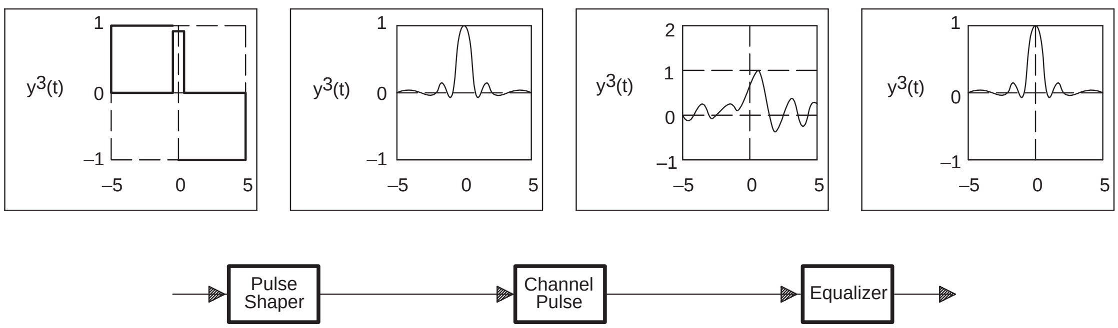
It is common practice to filter the signal pulses at the transmitter with the frequency characteristic described above in Figure 8. Having performed this operation, if the signals are sent down an ideal channel (that is, a channel with a channel-impulse response of δ function and no noise), the received signal should exhibit no ISI. Note that in general this condition will not be met, as the channel will have its own shaping effect on the transmitted signals. The effects of noise are considered in the next section of this paper.

In summary, the obvious problem caused by ISI is uncertainty in the received data samples. Instead of receiving the discrete levels that were transmitted, the receiver finds a continuous signal whose samples can take on any value. The receiver must then form an estimate from the received values to decide on the transmitted signal.

Equalization

As discussed in the *Pulse Shaping* subsection on page 155, a properly shaped transmit pulse resembles a sinc function, and direct superposition of these pulses results in no ISI at properly selected sample points. In practice, however, the received pulse response is distorted in the transmission process and may be combined with additive noise. Because the raised cosine pulses are distorted in the time domain, you may find that the received signal exhibits ISI. If you can define the channel impulse response, you can implement an inverse filter to counter its ill effect. This is the job of the equalizer. See Figure 9 below, which depicts the response to a single transmit pulse at various points in the system.

Figure 9. Transmission Process With Example Pulse Responses

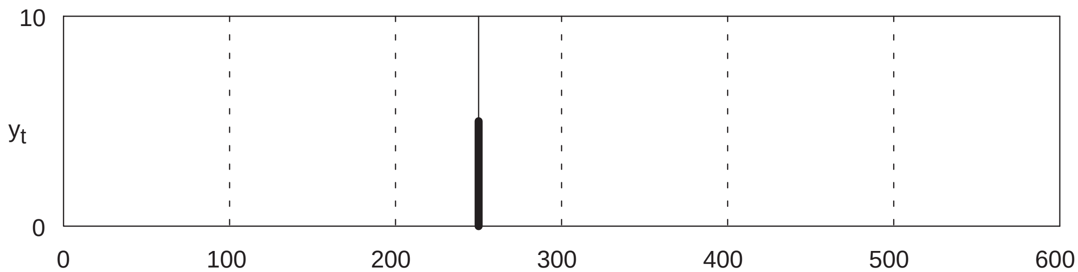


The original rectangular transmit pulse is shaped by the raised cosine filter. This ensures that the sampled spectra do not alias and therefore there is no ISI. The next waveform portrays the distorted impulse response received at the input of the equalizer. This distortion can be caused by spectral shaping due to a nonflat frequency response or multipath reception of the channel. This distortion can be removed by applying a filter that is the exact inverse (multiplicative inverse in spectral domain) of the channel frequency response.

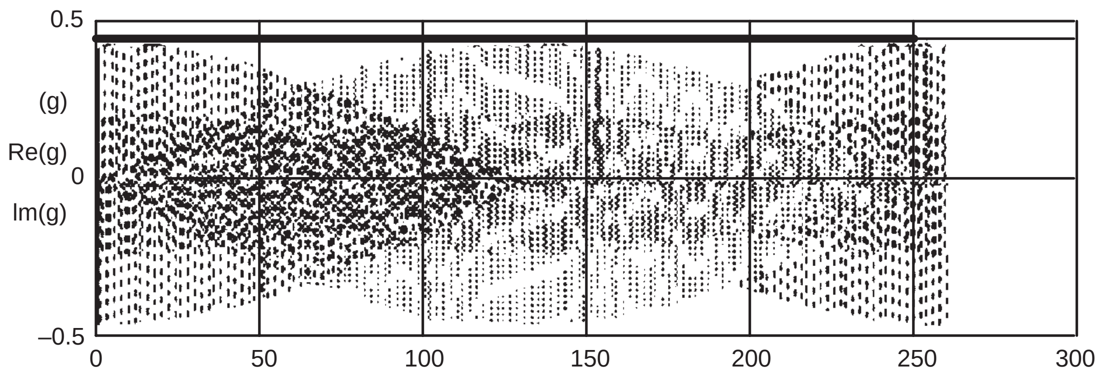
Multipath Effects on Frequency Response

Multipath effects describe the situation in which there are several propagation paths from transmitter to receiver. Most commonly, this results when there are reflected signals detected at the receiver following the direct path. The multipath phenomenon can be modeled by an FIR system. The center tap represents the direct path, while the succeeding tap weights represent the amplitudes, delays, and phases of the reflected paths. What does this look like in the spectral domain? For simple examples, see the two cases described in Figure 10 and Figure 11.

Figure 10. Case 1: Ideal Channel, No Multipath Effects



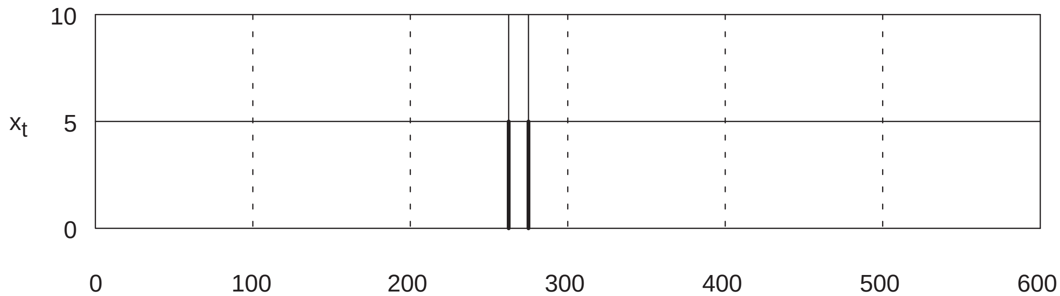
(a)



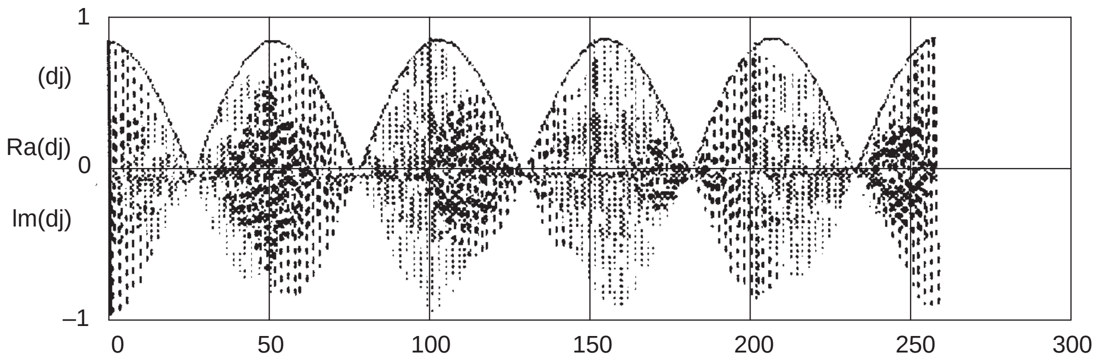
(b)

Figure 10(a) above shows the time response of an ideal transmission path, which is a δ function. Such a channel exerts no spectral distortion or delayed signals. Figure 10(b) shows the spectral response of such a system. Note that the frequency magnitude response is perfectly flat, as indicated by the solid horizontal line.

Figure 11. Case 2: System With a Single Unattenuated Multipath Channel



(a)

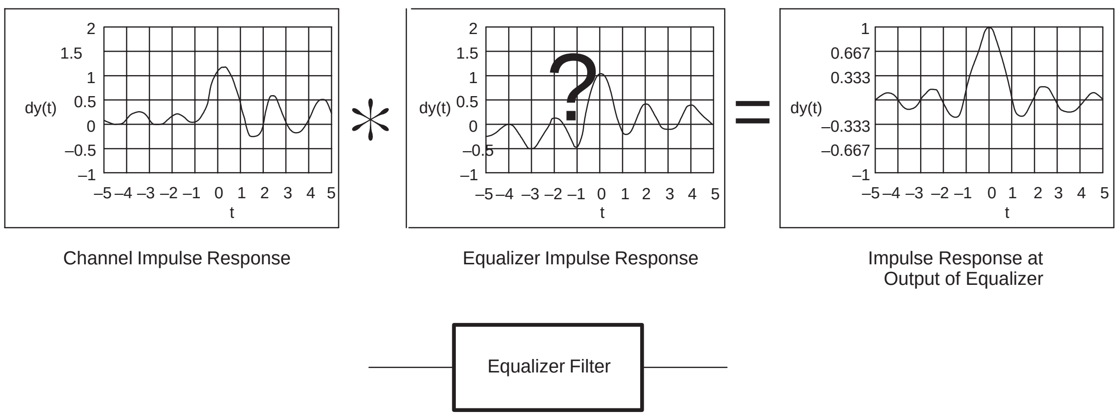


(b)

Figure 11(a) shows the time response of a system that contains a single multipath channel. The first nonzero sample of the response represents the direct path, while the second represents a delayed path to the receiver. In this instance, the pulses are identical in amplitude and phase and are separated by ten sample intervals. Notice in Figure 11(b) that the magnitude response exhibits $t_0/2$ nulls, where t_0 represents the sample delay. Even though you are effectively adding two identical flat spectra (as shown in Figure 10(b)), the time delay results in a phase delay in the spectral domain. This phase delay results in nulls where the two signals are of equal amplitude but opposite phase.

Obviously, multipath effects can have major effects on the system spectral response, thereby providing another justification for channel equalization.

Figure 12. Equalization Process



As depicted in Figure 12, the task of the equalization system is to determine and apply a filter that results in an equalized impulse response having zero ISI and channel distortion. This means that convolution of the channel impulse response and the equalizer impulse response must equal 1 at the center tap and have nulls at the other sample points within the filter span.

Two main techniques are employed to formulate the filter coefficients: *automatic synthesis* and *adaptation*. In automatic-synthesis methods, the equalizer typically compares a received time-domain reference signal to a stored copy of the undistorted training signal. By comparing the two, a time-domain error signal is determined that may be used to calculate the coefficient of an inverse filter. The formulation of this inverse filter may be accomplished strictly in the time domain, as is done in ZFE and LMS systems, which are examined in more detail in following sections. Other methods involve conversion of the received training signal to a spectral representation. A spectral inverse response can then be calculated to compensate for the channel response. This inverse spectrum is then converted back to a time-domain representation so that filter tap weights may be extracted.

The second method of filter synthesis is adaptation. In adaptation the equalizer attempts to minimize an error signal based on the difference between the output of the equalizer z_k and the *estimate* of the transmitted signal $\hat{x}_k$, which is generated by a decision device. In other words, the equalizer filter outputs a sample. The predictor or decision device determines what value was most likely transmitted. The adaptation logic endeavors to keep the difference between the two small. The main idea is that the receiver takes advantage of the knowledge of the discrete levels possible in the transmitted pulses. When the decision device quantizes the equalizer output, it is essentially throwing away received noise.

The main drawback of automatic synthesis is the overhead associated with the transmission of a training signal, which must be at least as long as the filter tap length. Typically, training is used to converge a filter at startup as part of the initialization overhead. Adaptation techniques can then be employed to track and compensate for minor variations in channel response on the fly [1].

Zero-Forcing Equalization

One computationally efficient method of forming an inverse filter is the *zero-forcing* technique. To formulate a set of FIR inverse filter coefficients, a training signal consisting of an impulse is transmitted over the channel. By solving a set of simultaneous equations based on the the received sample values, a set of coefficients can be determined to force all but the center tap of the filtered response to 0. This means the $N-1$ samples surrounding the center tap will not contribute ISI. The main advantage of this technique is that the solution to the set of equations is reduced to a simple matrix inversion.

The major drawback of ZFE is that the channel response may often exhibit attenuation at high frequencies around one-half the sampling rate (the folding frequency). Since the ZFE is simply an inverse filter, it applies high gain to these upper frequencies, which tends to exaggerate noise. A second problem is that the training signal, an impulse, is inherently a low-energy signal, which results in a much lower received signal-to-noise ratio than could be provided by other training signal types [1, 4].

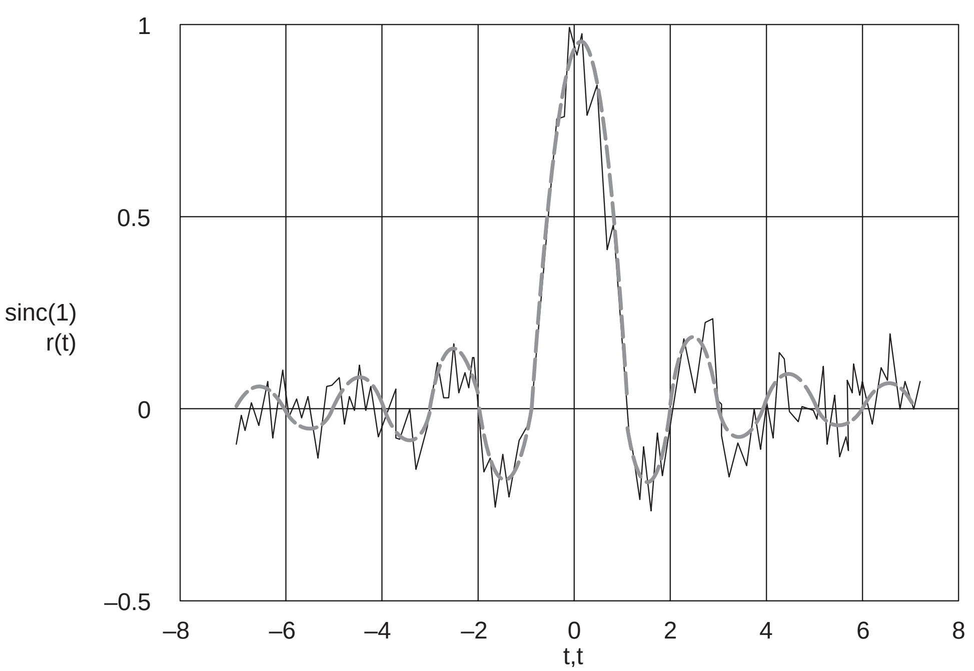
Example of 7-Tap ZFE Computation

First, create a simulated received pulse response. Begin with the equation of a sinc function, which is a simplification of the raised cosine pulse. Then simulate additive noise by the addition of random thermal *noise*. Finally, simulate sampling phase jitter with the random *jitter* term added to the time index. The simulated pulse response is plotted in Figure 13. The dotted trace represents the ideal noiseless channel response.

Figure 13. Simulated Pulse Response

$$sinc(t) := \frac{sin(\pi \cdot t)}{\pi \cdot t} \qquad noise(t) := rnd(0.2) - 0.1 \qquad jitter(t) := rnd(0.10) - 0.05$$

$$t := -7, -6.9,\ 7.0 \qquad r(t) := sinc(t + jitter(t)) + noise\ (t)$$



A vector is formed from the received samples. 2N − 1 samples are required to implement an N-tap filter. For the example 7-tap ZFE, you must collect 13 samples. Therefore, 13 equally spaced samples of r(t) are formed into the column vector V.

$$i := 0 \ldots 12 \qquad v_i := r(i-6)$$

$$v = \begin{bmatrix} 0.083 \\ -0.032 \\ 0.042 \\ 0.026 \\ -0.072 \\ 0.002 \\ 0.915 \\ 0.011 \\ -0.105 \\ 0.031 \\ -0.076 \\ 0.002 \\ 0.04 \end{bmatrix} \tag{15}$$

Next, form a matrix, PR, from the received samples. Each row consists of seven adjacent samples in time-reversed order. The first element of the top row is the center tap of the pulse response. The first element of the second row is the sample following the center tap, etc.

$$i := 0 \ldots 6 \quad PR_{0,i} := v_{6-i} \quad PR_{1,i} := v_{7-i} \quad PR_{2,i} := v_{8-i}$$

$$PR_{3,i} := v_{0-i} \quad PR_{4,i} := v_{10-i} \quad PR_{5,i} := v_{11-i} \quad PR_{6,i} := v_{12-i}$$

$$PR = \begin{bmatrix} 0.915 & 0.002 & -0.072 & 0.026 & 0.042 & -0.032 & 0.083 \\ 0.011 & 0.915 & 0.002 & -0.072 & 0.026 & 0.042 & -0.032 \\ -0.105 & 0.011 & 0.915 & 0.002 & -0.072 & 0.026 & 0.042 \\ 0.031 & -0.105 & 0.011 & 0.915 & 0.002 & -0.072 & 0.026 \\ -0.076 & 0.031 & -0.105 & 0.011 & 0.915 & 0.002 & -0.072 \\ 0.002 & -0.076 & 0.031 & -0.105 & 0.011 & 0.915 & 0.002 \\ 0.04 & 0.002 & -0.076 & 0.031 & -0.105 & 0.011 & 0.915 \end{bmatrix} \tag{16}$$

Next, compute the inverse of the channel response matrix PREQ.

$$PREQ := PR^{-1} \qquad PREQ = \begin{bmatrix} 1.101 & -9.62\times10^{-4} & 0.07 & -0.023 & -0.057 & 0.036 & -0.106 \\ -0.02 & 1.099 & -0.004 & 0.08 & -0.026 & -0.045 & 0.036 \\ 0.136 & -0.019 & 1.108 & -0.01 & 0.075 & -0.026 & -0.057 \\ -0.042 & 0.135 & -0.023 & 1.114 & -0.01 & 0.08 & -0.023 \\ 0.107 & -0.043 & 0.141 & -0.023 & 1.108 & -0.004 & 0.07 \\ -0.015 & 0.108 & -0.043 & 0.135 & -0.019 & 1.099 & -9.62\times10^{-4} \\ -0.023 & -0.015 & 0.107 & -0.042 & 0.136 & -0.02 & 1.101 \end{bmatrix} \quad \textbf{(17)}$$

The center column of PREQ contains the coefficients of the ZFE.

$$j := 0 \ldots 6$$

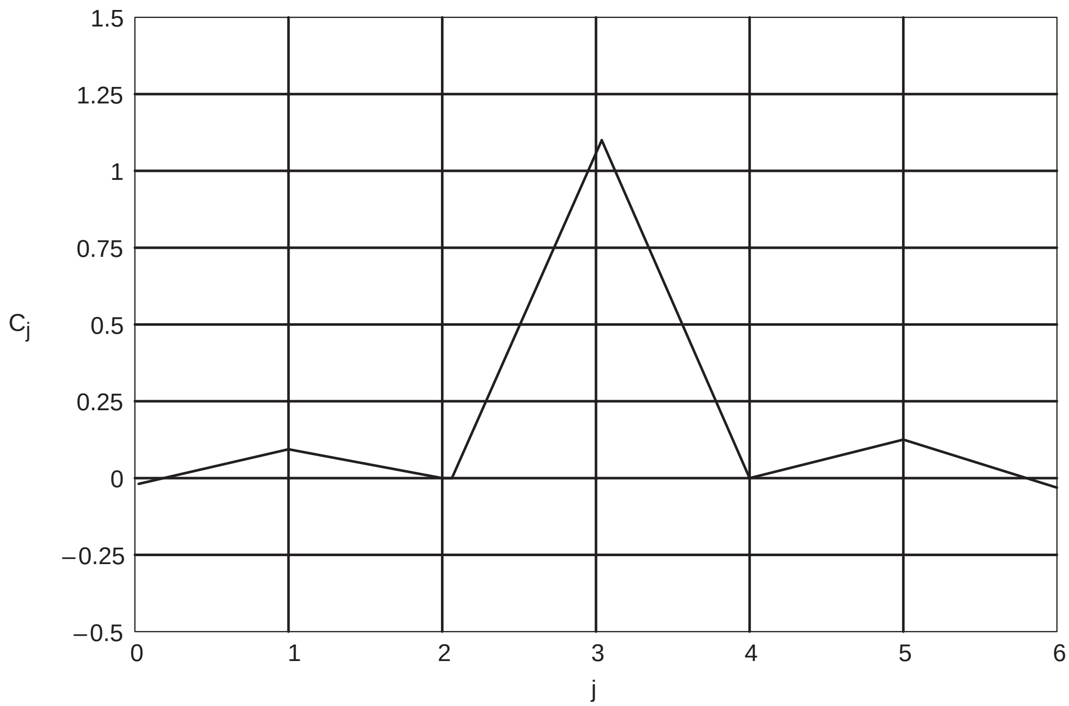
$$C_j := PREQ_{j,3} \qquad C = \begin{bmatrix} -0.023 \\ 0.08 \\ -0.01 \\ 1.114 \\ -0.023 \\ 0.135 \\ -0.042 \end{bmatrix} \quad \textbf{(18)}$$

Check the results by multiplying the coefficient vector by the row vectors of the received sample matrix. The dot products should result in the ideal channel response for the filter span, that is, 0, 0, 0, 1, 0, 0, 0. As shown below, the results check.

$$\begin{array}{lll} ROW0_j := PR_{0,j} & z := ROW0 \bullet C & z = 0 \\ ROW1_j := PR_{1,j} & z := ROW1 \bullet C & z = 0 \\ ROW2_j := PR_{2,j} & z := ROW2 \bullet C & z = 0 \\ ROW3_j := PR_{3,j} & z := ROW3 \bullet C & z = 1 \\ ROW4_j := PR_{4,j} & z := ROW4 \bullet C & z = 0 \\ ROW5_j := PR_{5,j} & z := ROW5 \bullet C & z = 0 \\ ROW6_j := PR_{6,j} & z := ROW6 \bullet C & z = 0 \end{array} \quad \textbf{(19)}$$

The coefficients for the ZFE filter response are shown plotted in Figure 14.

Figure 14. ZFE Filter Coefficient



Because the Japanese television broadcasters employed an impulse-like training signal, many of the first video deghosters for use in Japan employed ZFEs. To provide a higher signal-to-noise ratio (SNR) for the received training signal, these systems averaged the training signal over several training intervals. To further improve SNR, the U.S. broadcast industry has selected a chirp-like training signal, which has inherently higher energy. This signal, transmitted during the vertical blanking interval, allows suitably equipped receivers to automatically synthesize filters to alleviate the effects of multipath interference; that is, visible *ghost* images.

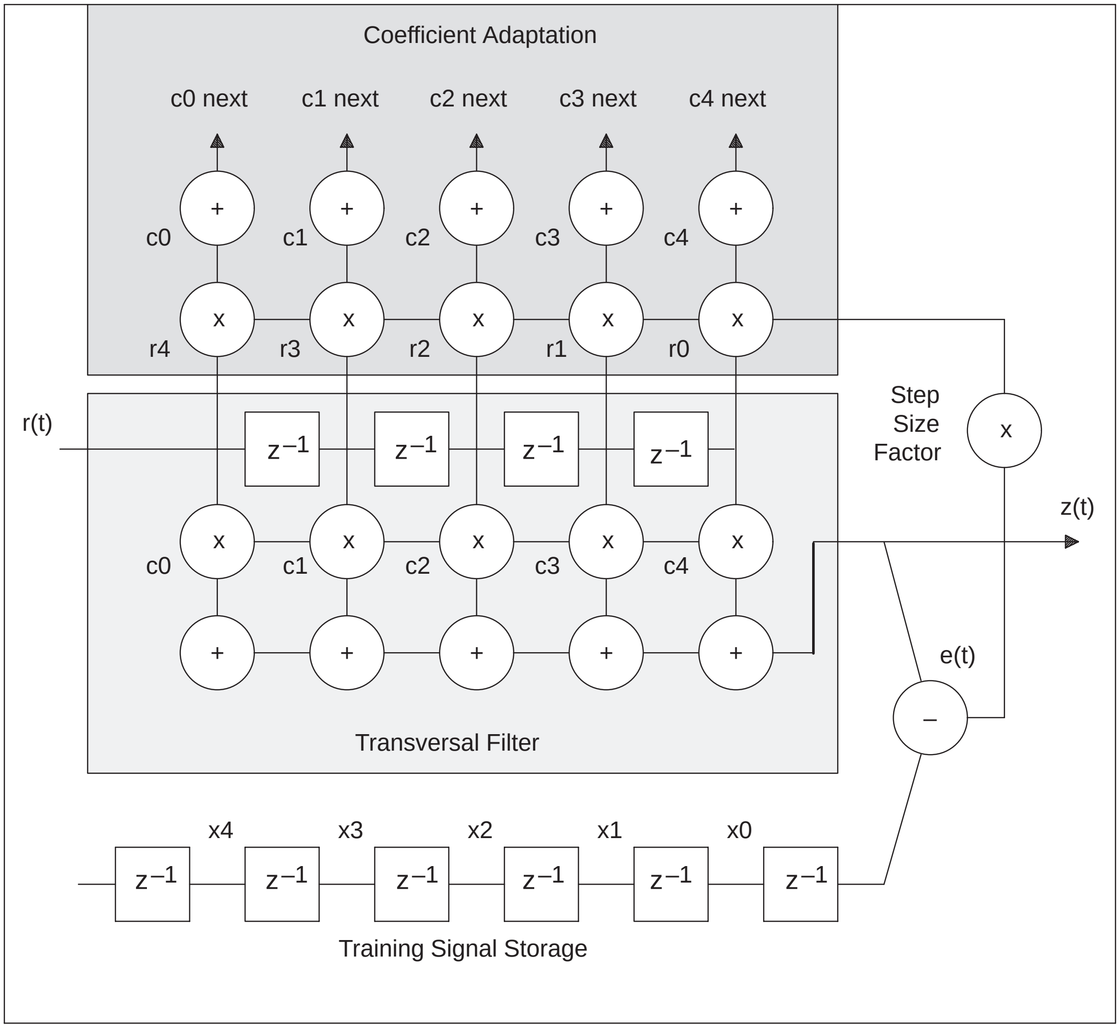
LMS Equalization

The least mean squared (LMS) equalizer is a more general approach to automatic synthesis. Instead of solving a set of N simultaneous equations as was done in the ZFE, the coefficients are gradually adjusted to converge to a filter that minimizes the error between the equalized signal and the stored reference. The filter convergence is based on approximations to a gradient calculation of the quadratic equation representing the mean square error. The beauty of the approach is that the only parameter to be adjusted is the adaptation step size αa. Through an iterative process, all filter tap weights are adjusted during each sample period in the training sequence. Eventually, the filter will reach a configuration that minimizes the mean square error between the equalized signal and the stored reference. As might be expected, the choice of αa involves a tradeoff between rapid convergence and residual steady-state error. A too-large setting for αa can result in a system that converges rapidly on start-up, but then chops around the optimal coefficient settings at steady state.

The LMS equalizer can also be shown to have better noise performance than the ZFE. Heuristically, the ZFE calculates coefficients based upon the received samples of one training signal. Since the captured data will always contain some noise, the calculated coefficients will be noisy — noise in / noise out. On the other hand, the LMS algorithm gradually adapts a filter based on many cycles of the training signal. If the noise is zero mean and is averaged over time, its effect will be minimized — noise integrates to 0.

A second major benefit to this approach is that you can employ any arbitrary training sequence. In general, you would prefer to use a high-energy signal to improve the received signal-to-noise ratio of the training sequence. In contrast, the unit impulse training signal required by the ZFE is probably the lowest energy flat-spectrum signal possible. Typical training sequences employed for LMS equalization include pseudorandom noise sequences and chirp-type signals.

Figure 15. Filter Output Computation



In Figure 15, the portion in the lower shaded rectangle is a standard transversal filter (FIR). The lower set of delays represents storage for the reference version of the training signal. Each time a sample is received, a new filter output is computed and compared to the corresponding reference signal, thereby forming an error signal. This error signal is then used to scale the received sample values contained in the filter storage elements. These scaled sample values are then added to the current filter coefficients to form the updated coefficients to be used at the next sample time.

The coefficients are updated according to the following equation:

$$C_n(k+1) = C_n(k) - \alpha e_k r_{k-n}, n = 0, 1..., N-1 \quad \textbf{(20)}$$

As an example, consider the calculation for the third tap weight (n = 2) at time k = 5:

$$C_2(5) = C_2(4) - \alpha e_4 r_2 \quad \textbf{(21)}$$

This means that the C_2 coefficient for the next sample period equals the current C_2 coefficient minus a correction term. The correction term is simply the current input sample corresponding to the C_2 tap multiplied by the current error value scaled by the adaptation rate term α.

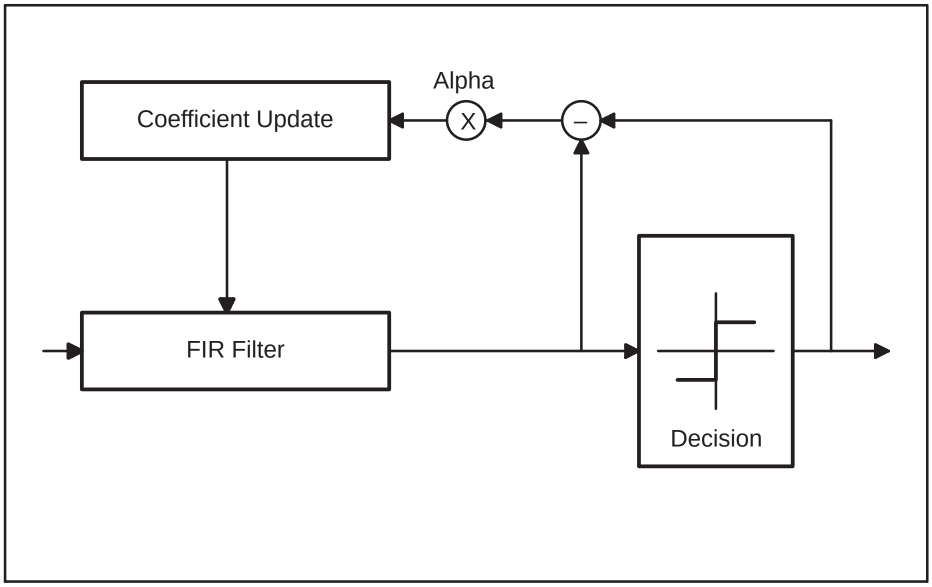
If the filtered sample output is much smaller than the actual value for the training signal, the error is a large negative value. The received sample values are scaled by this relatively large value, and the product is used to adjust the individual coefficients up or down (depending on the sign of the stored sample values) by a relatively large amount. For a smaller discrepancy between the filter output and training signal, the error sample and, hence, the amount of adjustment, will be smaller. The fact that each coefficient is changed by a different adjustment term (based on distinct received samples) allows the filter coefficients to converge from any initial state to one that minimizes the mean square error between the received training signal and the reference.

Decision-Directed Equalization

The previous equalizer systems are linear in that they employ linear transversal filter structures. The filters implement a convolution sum of a computed impulse response with the input sequence. Often with data communication systems, one can take advantage of prior knowledge of the transmit signal characteristics to deduce a more accurate representation of the transmit signal than can be afforded by the linear filter. For instance, a bipolar transmit signal consists of pulses with amplitudes of ± 1. This signal is then pulse shaped, distorted by the analog channel, and filtered by a linear FIR filter. The processed signal is no longer a bipolar sequence. Instead, the output values span the range of values representable by the hardware, for example, the range of numbers specified by Q15 notation [5]. It is possible to devise a *decision device* (a predictor or a slicer) that estimates what symbol value was most likely transmitted, based on the linear filter continuous output. For example, in the case of the bipolar sequence transmission scheme, a very simple decision device could replace all positive values with a positive 1 and all negative values with a negative 1. The difference between the decision device input and output forms an error term which can then be minimized to adapt the filter coefficients. This is true because a perfectly adapted filter would produce the actual transmitted symbol values, and, therefore, the slicer error term would go to 0. In practice, the error is never 0, but if the adapted filter is near ideal, the decisions are perfect. In this case, the slicer is effectively throwing away received noise with each decision made.

Coefficients can be updated in a manner similar to that employed by the LMS equalizer. There is, however, one important distinction. In Figure 16, the error term is computed as the difference between the input and the output of the decision device, as opposed to the LMS error term, which was based on a stored reference training signal. This means that the decision-directed equalizers do not require a training sequence. This is a major distinction between automatic synthesis (which requires a training signal) and adaptive techniques (which do not require a training signal).

Figure 16. Decision-Directed Equalization



Decision-Feedback Equalization

Another nonlinear adaptive equalizer should be considered: the decision feedback equalization (DFE). DFE is based on the principle that once you have determined the value of the current transmitted symbol, you can exactly remove the ISI contribution of that symbol to future received symbols (see Figure 17). The nonlinear feature is again due to the decision device, which attempts to determine which symbol of a set of discrete levels was actually transmitted. Once the current symbol has been decided, the filter structure can calculate the ISI effect it would tend to have on subsequent received symbols and compensate the input to the decision device for the next samples. This postcursor ISI removal is accomplished by the use of a feedback filter structure.

Figure 17. Received Signal Including Additive Noise Effects

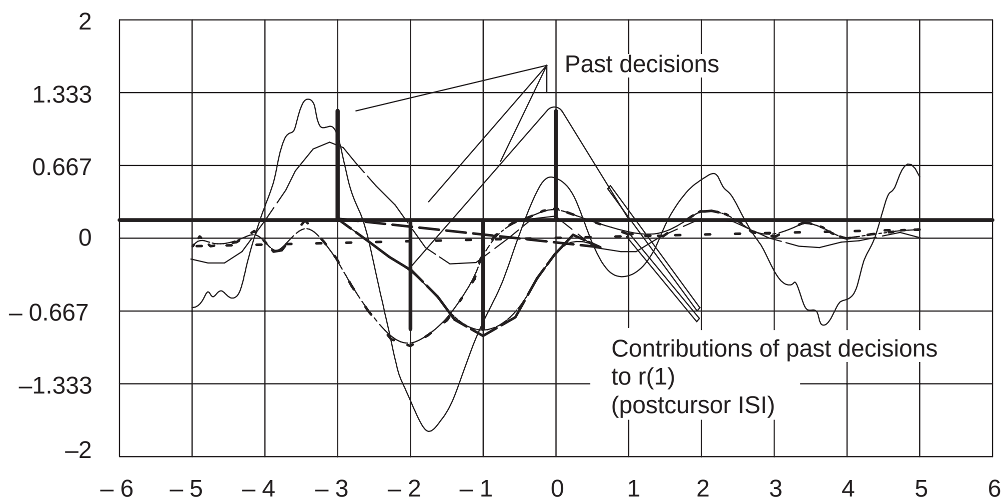
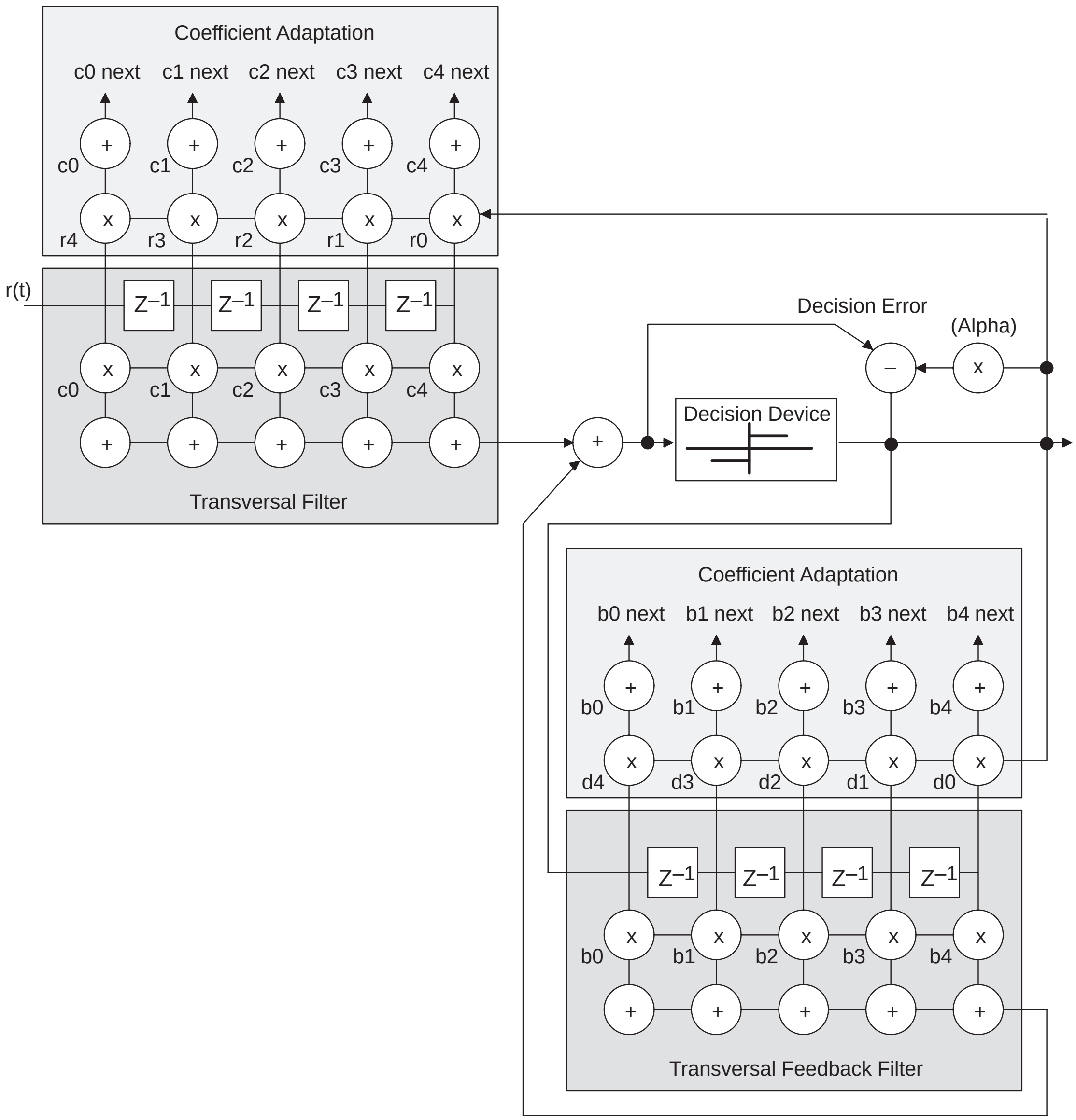


Figure 17 shows the received signal, including the effects of additive noise. Superimposed are the past four decisions (r(0), r(–1), r(–2), r(–3)) and the traces corresponding to the channel response for each pulse. Because you have a transverse filter that mimics the the system response, you can subtract the ISI contributions of the past symbols (as decided) from the next received symbol (t = 1). You can see that the decision values are sliced to ± 1, thereby tossing away noise that would have otherwise improperly influenced the compensation for the postcursor ISI. As shown in Figure 17, you can see that the coefficients of the feedback filter should converge to the right half of the channel impulse response. That is because the output value at any time t consists of the current sample times the center tap weight, plus the previous samples times the right half of the impulse response, plus the subsequent samples times the left half of the impulse response. It is the previous samples times the right half impulse response that will be subtracted by the feedback filter.

Figure 18. DFE Functional Block Diagram



In Figure 18, you can see that the DFE contains all of the same functional blocks as the previously described decision-directed equalizer. In addition, there is a second adaptive filter structure fed by the output of the decision device. This second filter is the feedback stage that cancels the postcursor ISI. Its inputs are the symbol decisions, and the tap weights converge through the LMS process to resemble the tail of the channel impulse response (taps beyond the center tap).

The adaptation formula for the feedback tap coefficients can be the same as for the feed forward section. For the LMS approximation [1, 4]:

$$c_n(k+1) = c_n(k) - \alpha e_k r_{k-n}, n = 0, 1 \ldots, N-1 \tag{21}$$

$$b_n(k+1) = b_n(k) - \alpha e_k d_{k-n}, n = 0, 1 \ldots, N-1 \tag{22}$$

Adaptive Equalization for Digital Cellular Telephony

The direct sequence spreading employed by CDMA (IS-95) obviates the need for a traditional equalizer. The TDMA systems (for example, GSM and IS-54), on the other hand, make great use of equalization to contend with the effects of multipath-induced fading, ISI due to channel spreading, additive received noise, and channel-induced spectral distortion, etc. Because the RF channel often exhibits spectral nulls, the linear equalizers are not optimal due to their tendency to boost noise at the null frequencies. Of the nonlinear equalizers, the DFE is currently the most practical system to implement in a consumer system. As discussed below, there are other designs that outperform the DFE in terms of convergence or noise performance, but these generally come at the expense of greatly increased system complexity. Today, most TDMA phones employ DFE running on fixed-point DSPs such as those in the TMS320C5x [6] family. For a detailed look at some representative systems, consult *A Low-Effort DSP Equalization Algorithm for Wideband Digital TDMA Mobile Radio Receivers* [7] and *Channel Equalizer for a Digital Mobile Telephone Using Narrow-Band TDMA Transmission* [8].

Advanced Adaptive Equalizer Structures

Several adaptation schemes and alternate filter structures offer better performance in some respects than those described above. Usually this performance improvement comes at the cost of increased complexity in terms of DSP CPU loading or logic gate count. For the most part, these are well understood algorithms whose system performance is still being evaluated in various applications. In any case, their treatment is beyond the scope of this tutorial in equalization concepts, and references are cited on page 174 for the interested reader.

Lattice Filter Structures

In general, the well-known lattice filter structure [9] can be substituted for the FIR sections in the DFE system. The lattice DFE has been shown to be less sensitive to roundoff errors than the transverse filter DFE, though it has comparable convergence properties. Special forms of LMS and RLS adaptation for lattice structures are summarized in *Adaptive Equalization for TDMA Digital Mobile Radio* [10]. For a detailed discussion of the implementation of lattice DFE for digital cellular radio, refer to *An Adaptive Lattice Decision Feedback Equalizer for Digital Cellular Radio* [11].

RLS Adaptation

RLS adaptation refers to the recursive least squares algorithm. The RLS algorithm can be designed to converge significantly faster than the LMS technique converges. Recall that the LMS coefficients are adjusted during each sample period by the product of the current error multiplied by the appropriate signal sample scaled by α. In the case of RLS, the adaptation is similar, but instead of scaling the adjustment by α, a value derived from the inverse of the sample autocorrelation matrix is used to scale the error/sample product. As a comparison of complexity, a 20-tap (10 forward, 10 feedback) LMS update system requires about 40 operations. A standard RLS update, on the other hand, requires on the order of 1000 operations [10]. For a more detailed look at RLS in digital cellular systems, see *A Decision Feedback Equalizer With a Frequency Offset Compensating Circuit for Digital Cellular Radio* [12] and *Bidirectional Equalization Technique for TDMA Communication Systems Over Land Mobile Radio Channels* [13].

Probabilistic Detection Algorithms

Two more advanced adaptation techniques that employ stochastic principles to minimize the probability of error are *maximum a posteriori probability* (MAP) and *maximum likelihood sequence estimation* (MLSE). These techniques require knowledge of the channel characteristics and the probability distribution of the additive noise. MAP is a symbol-by-symbol detector; whereas, the MLSE algorithm employs the Viterbi algorithm (VA) to minimize the probability of sequence error. Both approaches provide comparable performance and are still regarded as prohibitively complex for channels with a long impulse response, because complexity is exponentially related to the ISI span. For a further study, consult references [10] and [14].

Code Availability

The associated program files are available from the Texas Instruments TMS320 Bulletin Board System (BBS) at (713) 274-2323. Internet users can access the BBS via anonymous ftp at *ti.com.*

References

1. Qureshi, S.,“Adaptive Equalization”, *IEEE Communications Magazine*, March 1992, pp. 9–16.

2. Peebles, P.Z., *Communication System Principles,* Addison-Wesley, 1976.

3. Samueli, H., Daneshrad, B., Joshi, R., Wong, B., and Nicholas, H., “A 64-Tap CMOS Echo Canceller/Decision Feedback Equalizer for 2B1Q HDSL Transceivers”, *IEEE Journal on Selected Areas in Communications*, Vol. 9, Iss: 6 , August 1991, pp. 839–847.

4. Ziemer, R.E., and Peterson, R.L., *Introduction to Digital Communication*, Macmillan, 1992.

5. Lovrich, A. and Simar, R., “Implementation of FIR/IIR Filters with the TMS32010/TMS32020”, *Digital Signal Processing Applications with the TMS320 Family,* Volume 1, Texas Instruments, 1989.

6. *TMS320C5x User’s Guide,* Texas Instruments, 1993.

7. Bune, P., “A Low-Effort DSP Equalization Algorithm for Wideband Digital TDMA Mobile Radio Receivers”, *International Conference on Communications Conference Record*, June 1991, pp. 763–767.

8. Svensson, L., “Channel Equalizer for a Digital Mobile Telephone Using Narrow-Band TDMA Transmission”, *39th IEEE Vehicular Technology Conference,* Volume 1, 1989, pp. 155–158.

9. Oppenheim, A.V,. and Schafer, R.W., *Discrete Time Signal Processing*, Prentice-Hall, 1989.

10. Proakis, J.G., “Adaptive Equalization for TDMA Digital Mobile Radio”, *IEEE Transactions on Vehicular Technology,* Volume 40, No. 2, May 1991.

11. Narasimhan, A., Chennnakeshu, S., and Anderson, J.B., “An Adaptive Lattice Decision Feedback Equalizer for Digital Cellular Radio”, *40th IEEE Vehicular Technology Conference*, 1990, pp. 662–667.

12. Shimizaki, Y., Nakai, T., Ono, S., and Kondoh, N., "A Decision Feedback Equalizer With a Frequency Offset Compensating Circuit for Digital Cellular Radio", *Vehicular Technology Society 42nd VTS Conference*, Volume 2, 1992, pp. 596–599.

13. Liu, Yow-Jong, "Bidirectional Equalization Technique for TDMA Communication Systems Over Land Mobile Radio Channels", *Proceedings*, *GLOBECOM,* 1991.

14. Jung P. and Baier, P.W., "VLSI Implementation of Soft Output Viterbi Equalizers for Mobile Radio Applications", *Vehicular Technology Society 42nd VTS Conference,* Volume 2, 1992, pp. 577–85.

15. *Mathcad 4.0*, MathSoft Inc., 210 Broadway, Cambridge, MA 02139.