

Embedded Systems Design

feature

Long wires or distant wireless communications **always garble the transmission**. How can an engineer cure that distortion? Turns out, **a lot of math and a little experience can work wonders**. This author provides **both, giving us priceless recipes for cleaning up long-range communications**.

Curing nonlinear distortion

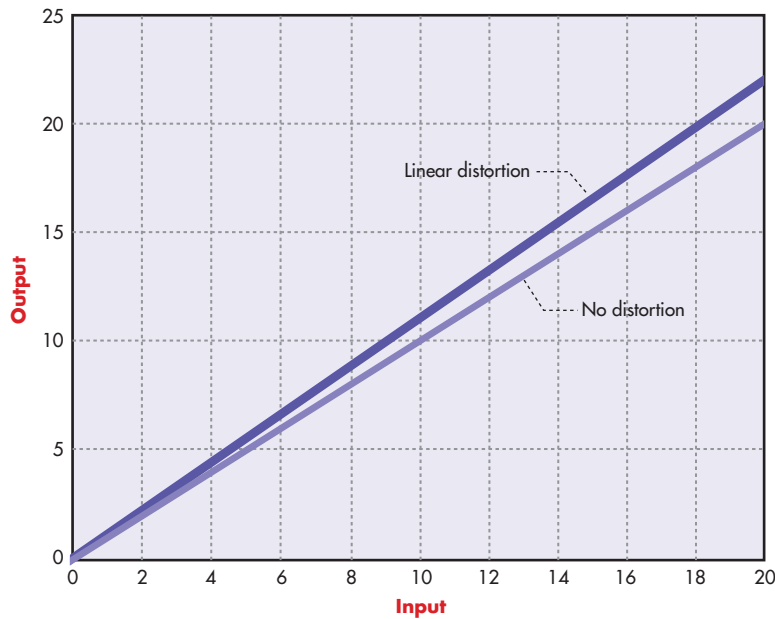
BY ROY BATRUNI

Nonlinear distortion is a persistent problem that can severely reduce a communications channel's capacity to carry information. Although *linear* distortion is more common, it's simpler to counteract with linear signal-processing techniques. In contrast, *nonlinear* distortion comes from functions that are higher-order and more-complex than those in simple linear distortion. Whereas linear distortion always generates signal-dependent components at the same frequencies as the input signal, nonlinear distortion results in signal-dependent distortion components, or harmonics, which are at frequencies that weren't present in the input signal. For these reasons nonlinear distortion is extremely complex to model and even more complex to correct. The harmonics from nonlinear distortion cause the bandwidth of the output signal to be wider than the channel input (which isn't the case with a channel affected by linear distortion). This causes many problems in telecommunications, where adjacent-channel signal rejection is an important issue.

In this article, I'll describe what causes nonlinear distortion and how you can avoid it in your communications designs.



Linear distortion: the transfer characteristics between x and y is a straight line



The lower line represents a 1:1 ratio input-to-output transfer characteristic, which means there is no distortion in going from input signal to output signal. The top line represents a 1:1.1 ratio input-to-output transfer characteristic, which means there is distortion in going from input signal to output signal, and since this transfer characteristic is a straight line, the distortion is referred to as linear distortion. The two lines differ in their slopes or inclination; the lower line has a slope of 1 and the top line a slope of 1.1.

Figure 1

Three-dimensional example of linear distortion; the output as a function of the two inputs is a plane

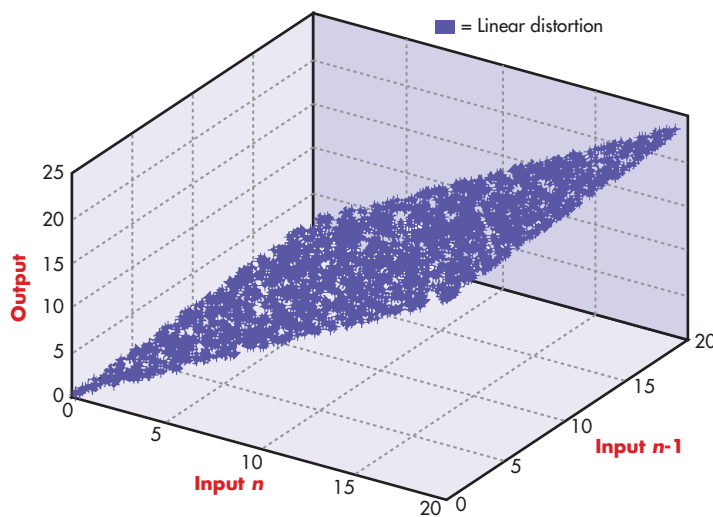


Figure 2

IS IT NOISE OR DISTORTION?

We can group impairments to signal fidelity into two general categories: noise and distortion. Noise is normally an external interference that corrupts the signal, where “external” means that noise sources are al-

most always independent of the signal itself. Distortion, on the other hand, is almost always a function of the signal itself, which means that the level and severity of the signal corruption can be large or small depending on the signal dynamics. Those

dynamics can include signal level, phase, power, slew rate, and so on.

Because it’s external, noise is a problem that always places a limit on signal integrity. Once noise interferes with the signal it can’t be removed. There are techniques to improve a signal’s immunity to noise, but only to the extent that this immunity approaches an upper limit (known as the Shannon Limit).

Distortion is signal-induced; if you know the dynamics that cause the signal to be distorted, you can reverse them and subtract them from the signal to restore it to its original form. In practice, when noise is present it’s impossible to fully restore the signal to its original form because noise generates uncertainty as to the actual value of the signal. Still, even when there’s noise present, distortion can be greatly reduced.

From this point on I’ll ignore noise and focus only on distortion. As a self-induced deformation of the signal, distortion can only be removed by using the distorted signal itself. A reference undistorted version of the signal can be used when extracting the distortion properties. In real-time operation, however, this reference signal isn’t available (or there’d be no need to remove the distortion!), so we use a model of the distortion properties along with the distorted signal to generate a corrected signal.

LINEAR DISTORTION

We can model signal distortion using algebraic expressions. These can range from simple equations to very complicated ones covering linear and nonlinear distortion, respectively. When linear distortion is involved, the models are simple. Nonlinear distortion requires much more complex algebra to model and to remove from a distorted signal.

A simple form of signal distortion occurs when a fraction of the signal amplitude is summed with the signal itself. For example, a signal x is corrupted in such a way that it becomes $y=x+0.1x$. Or, rewriting as $y=1.1x$, the corrupted signal y is always a constant multiple of x . Clearly, if this simple distortion function is known, we can always divide y by 1.1 to obtain the original signal x . This is a very simple form of distortion referred to as *linear* distortion, so called because the transfer characteristic from x to y is a straight line. The transfer characteristic between x and y is illustrated

Plots compare no distortion, linear distortion, and nonlinear distortion

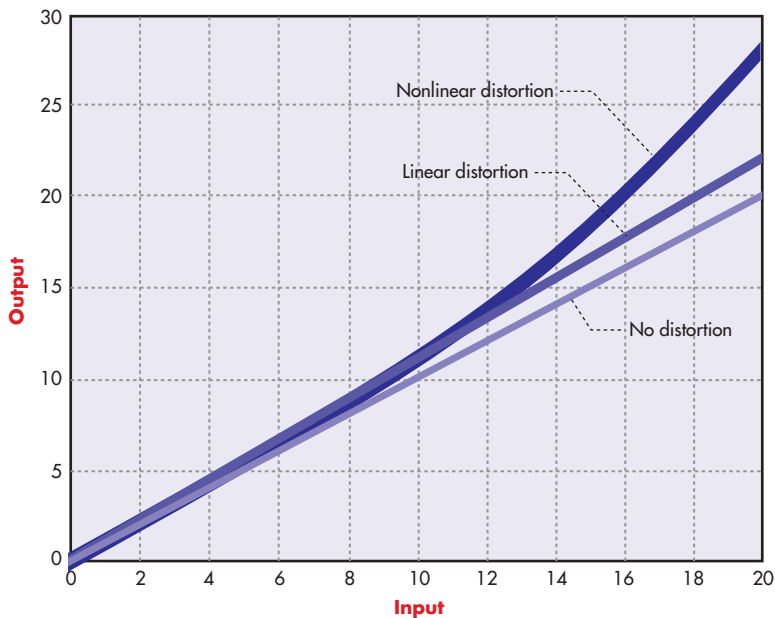


Figure 3

Three-dimensional plot of $y_n = x_n + 0.001x_{n-1}^3$, where the output at time n is a function of the signal at time n and a cubic power of the signal at time $n-1$; clearly the surface of this input-output transfer characteristic is not a plane, but a curved manifold

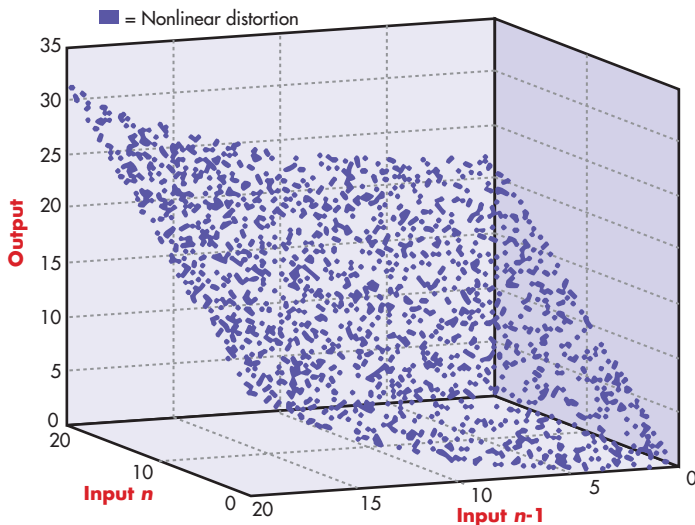


Figure 4

in Figure 1.

This simple example is two-dimensional (output signal vs. input signal) but the same arguments extend to higher dimensions. For example, in three dimensions, the signal history can be a factor in

the distortion such that the output is $y_n = 1.1x_n + 0.1x_{n-1}$ where n denotes the current time sample and $n-1$ the previous time sample. The three-dimensional transfer characteristic is shown in Figure 2, and the three dimensions are x_n , x_{n-1} , and y_n .

NONLINEAR DISTORTION

The simple examples of linear distortion can be made more complex by adding more terms to the equation such as $y_n = 1.1x_n + 0.1x_{n-1} - 0.01x_{n-2}$ provided that all terms are a function of the signal and its history multiplied by fixed numerical terms. If, on the other hand, one or more of those signal-history components is raised to a power other than one, for example:

$$y_n = 1.1x_n + 0.1x_{n-1}^3 - 0.01x_{n-2} \quad (1)$$

we have nonlinear distortion. The term *nonlinear* is used because the surface of the input-output transfer characteristic is no longer a straight line, plane, or hyperplane (a multidimensional plane). Figure 3 shows the plot of:

$$y_n = x_n + 0.001x_n^3 \quad (2)$$

in addition to the linear-distortion and no-distortion plots shown in Figure 1. A three-dimensional example using:

$$y_n = x_n + 0.001x_{n-1}^3 \quad (3)$$

is shown in Figure 4.

Nonlinear distortion can be even more complex than these examples show. However, even simple distortion components like the ones shown here can cause severe loss of signal fidelity. This loss is illustrated easily when we model the signal x_n as a sine wave.

Let $x_n = \sin(\omega nT)$ where T is the sampling period, and ω is the angular frequency of the signal, with nT denoting the sampling time. When linear distortion is present the output is

$y_n = x_n + 0.1x_{n-1} = \sin(\omega nT) + 0.1\sin(\omega(n-1)T)$ and this simply results in an output signal y_n that is a sine wave of the same frequency but is shifted in amplitude and phase, $y_n = \sin(\omega nT) + 0.1\sin(\omega(n-1)T) = a \sin(\omega(n-1)T + \theta)$. This means that linear distortion can only alter a sinusoidal input's amplitude and phase, and can't alter its frequency.

Now for a simple nonlinear distortion:

$$y_n = x_n + 0.1x_n^2 = \sin(\omega nT) + 0.1 \sin^2(\omega nT) \quad (4)$$

the second term that is raised to a power of

This sinusoid shown in a solid line is the input undistorted signal; the dotted line is the linearly distorted signal, which has the same frequency, only shifted in phase and amplitude

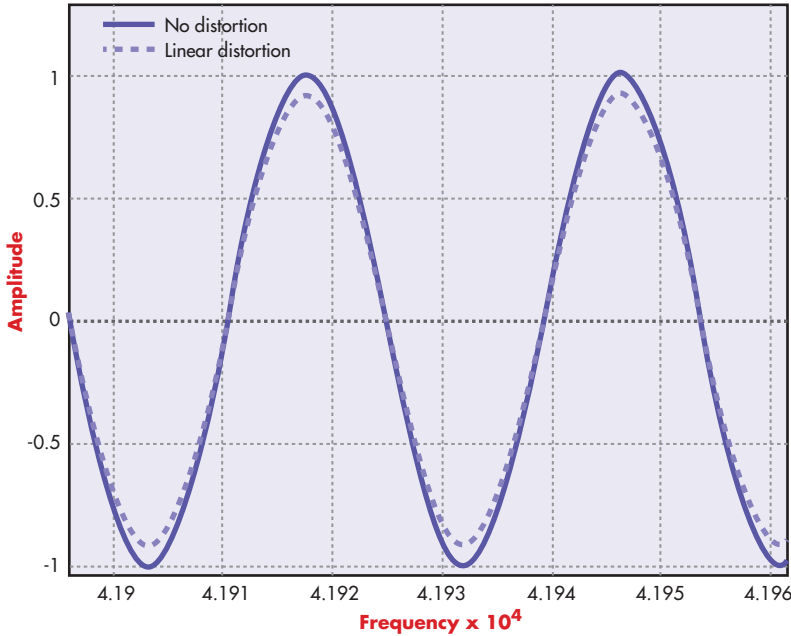


Figure 5

Input and output signal amplitude spectra are superposed in this Fourier transform plot

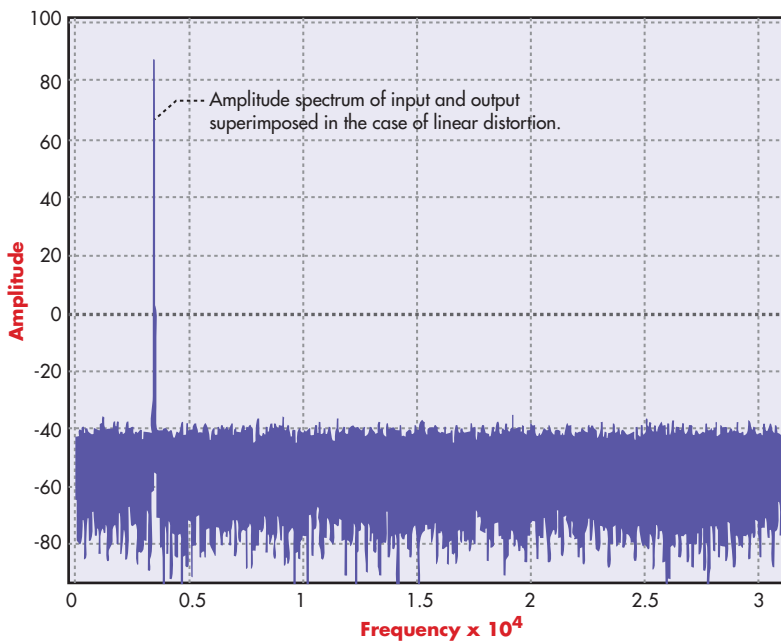


Figure 6

2 becomes $\sin(\omega n T)^2 = 0.5 - 0.5 \cos(2\omega n T)$ using simple trigonometric identities. The overall expression then becomes:

$$y_n = x_n + 0.1 x_n^2 = \sin(\omega n T) + 0.05 - 0.05 \cos(2\omega n T) \quad (5)$$

and the most serious component of the distortion now is that the output y_n has a cosine term that is twice the frequency of the input sinusoid. This modification of the input frequency doesn't occur in linear distortion.

In Figure 5 an example of linear distortion shows where the output sine wave is only shifted in amplitude and phase from the input signal. The frequency domain plot, known as the Fourier transform plot, is shown in Figure 6. The frequency component for the input and output signals is the same.

A nonlinear distortion example using the equation:

$$y_n = x_n - 0.001 x_n^3 \quad (6)$$

is shown in Figure 7, where the sinusoid is the input signal and the pseudo-sinusoid that has a higher frequency is the distortion component due to the cubic distortion term. The Fourier transform amplitude spectrum is shown in Figure 8 where the third harmonic (the frequency component at three times the input frequency) is clearly prominent.

CORRECTING FOR SIGNAL DISTORTION

Linear distortion changes the signal's amplitude and phase. In applications where those parameters are important, such as audio engineering, correcting for linear distortion is limited only by the amount of external noise interfering with the signal. The lower the noise power, the easier it becomes to correct for linear distortion. In the limit, if no noise is present, linear distortion can be completely removed. As an example, the transfer function $y_n = 1x_n + 0.1x_{n-1}$ can be completely corrected by the function $z_n = 1y_n - 0.1z_{n-1} + 0.01z_{n-2}$, and this results in $z_n = x_n$.

Nonlinear distortion results in a change in the signal's amplitude and phase as well as the generation of signal components at frequencies that weren't present in the input

Nonlinear distortion: input sinusoid and output distortion term at three times the frequency (amplified)

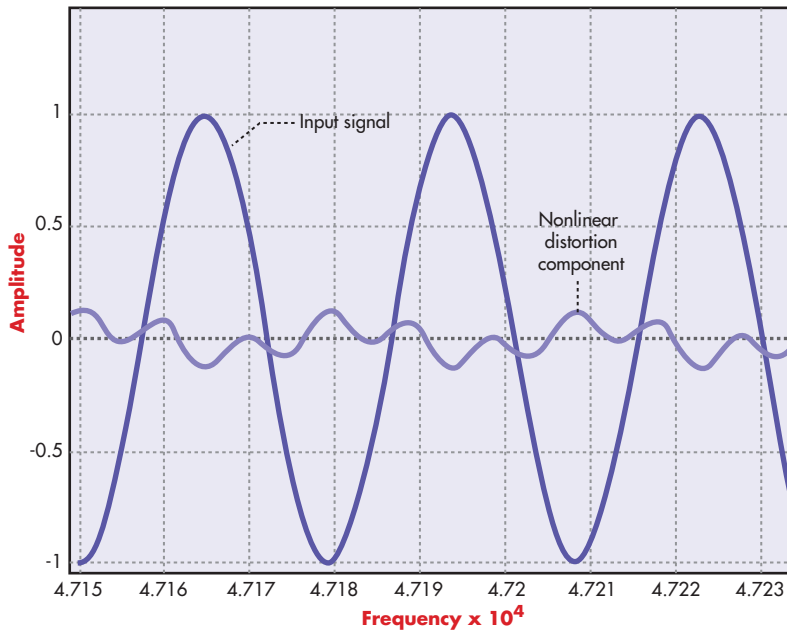


Figure 7

Fourier transform amplitude spectrum of the output signal with nonlinear distortion

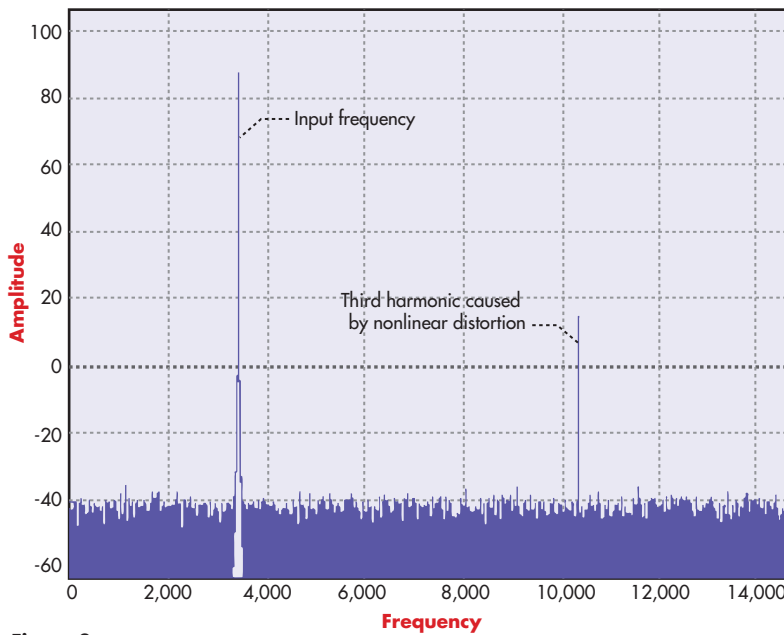


Figure 8

signal. This is often a more severe effect in real applications than linear distortion is. For example, in wireless communications the transmitted signal is usually allocated a certain amount of spectrum. Nonlinear distortion causes the received signal to spill over to adjacent frequencies, causing interference between different user channels.

Correcting for nonlinear distortion isn't simple, even in the absence of noise. A simple nonlinear function such as:

$$y_n = x_n + 0.1 x_n^3 \quad (7)$$

has an inverse with an infinite number of

terms. Practical inverse functions that are simple to implement are therefore only approximations of the real inverse.

NONLINEAR SIGNAL PROCESSING TECHNIQUES

Nonlinear signal analysis isn't a new science. There are highly complex formats for modeling different types and levels of severity of nonlinear distortion. The vast majority of those models involve either polynomial-based structures or functional-based kernels, with the consequences that a silicon- or software-based solution to the nonlinear distortion model is characterized by:

- High complexity—and therefore high cost—due to the complex computation of higher-power signal terms or complex functionals
- Sensitivity to model errors, in that a slight variation in model parameters results in large variation in the model output
- Susceptibility to noise-gain problems where higher-order signal terms in a polynomial model will result in higher noise powers and adverse changes in the noise statistics
- Impracticality of implementing adaptive solutions because of the cross-coupling of signal terms in polynomials during adaptation, lack of differentiability of functionals, or lack of existence of a global error-surface minimum
- The impracticality of obtaining an inverse of a nonlinear channel model since often a polynomial with a few higher-order terms requires an inverse with an infinite series of terms

Many applications can benefit from improved channel or signal linearity, but a few examples are satellite communications, magnetic recording, fiber-optic transmission, underwater acoustics, and almost all applications where analog electronics are used, such as data conversion and signal amplification.

STRUCTURES FOR MODELING NONLINEAR DISTORTION

Once a signal transfer function is allowed to have terms that are nonlinear, the combinations of powers, functions, and coefficients are endless. We'll therefore restrict our discussion to just a few possibilities to see how

Channel estimation generates a replica of the channel under test

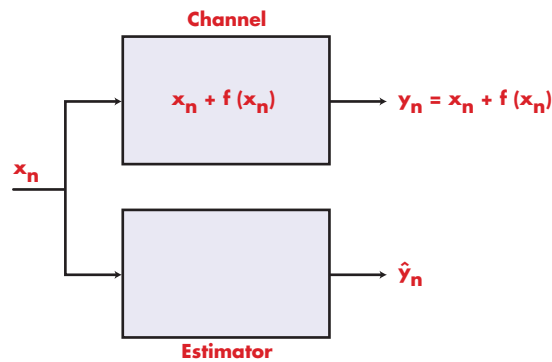


Figure 9

Channel inversion generates an inverse of the channel under test

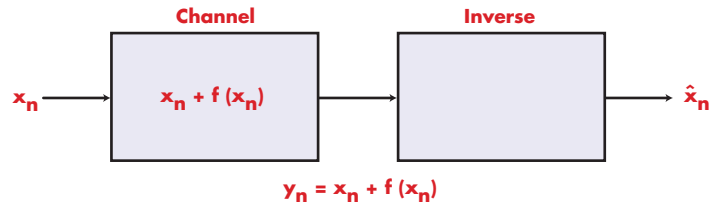


Figure 10

nonlinear signal processing applies to all types of nonlinearities.

For our purposes we'll group nonlinear functions into one or more of these categories:

- Memoryless distortion with generalized polynomial representation
- Memoryless distortion with orthogonal polynomial representation
- Nonlinear polynomial distortion model with memory effects
- Nonlinear functional distortion model with memory effects
- Discontinuous nonlinear models

In practice, any real system could have all of these nonlinear distortions plus other possibilities not listed here. To keep things manageable we'll discuss each of these categories in isolation.

MEMORYLESS DISTORTION WITH GENERALIZED POLYNOMIAL REPRESENTATION

The simplest form of nonlinear distortion is the "memoryless" distortion that can be modeled as a higher-order polynomial. Let x_n be an input signal to a channel that has the additive nonlinear distortion function:

$$f(x_n) = k_2 x_n^2 + k_3 x_n^3 + k_4 x_n^4 \quad (8)$$

then the output is:

$$y_n = x_n + f(x_n) = x_n + k_2 x_n^2 + k_3 x_n^3 + k_4 x_n^4 \quad (9)$$

The distortion is *memoryless* because $f(x_n)$ doesn't have any elements that are a function of time samples $n-1$, $n-2$, and so forth.

Engineers want to solve two basic problems when they encounter such a channel. The first is channel estimation; the second is channel inversion.

Channel estimation involves extracting a replica of the channel when its input and output are available for analysis, as shown in Figure 9. *Channel inversion* involves obtaining a transfer function that, when cascaded with the channel, results in an overall output equal to the original channel input, as shown in Figure 10.

When you're performing either estimation or inversion you usually don't know either the order of the polynomial or the value of the coefficients. Offline and real-time methods for estimation and inversion must therefore be able to estimate both the polynomial order and the coefficient values. This is where the first problem with nonlinear systems arises. If you over- or under-model the nonlinear polynomial order, the solutions to estimation and inversion will be flawed—and it's almost impossible to get an exact estimate in practice.

Still, assuming we do somehow magically determine the order, it's not straightforward to obtain the coefficient estimate because methods such as least squares can have local minima that lead to a suboptimal solution. When it comes to inversion things are even more difficult; an inverse of a polynomial usually involves fractional powers and/or an infinite series of terms, making such inverses impractical. One approximation that is often made is the so-

called *p*th-order inverse, where enough terms are computed in the inverse such that the first *p* terms in the distortion function are cancelled but higher-order terms remain (even some higher-order terms that were not present before are created).

Another practical and severe problem with *p*th-order inverses is that the higher-order terms created above the *p*th-order often take effect at larger amplitudes. So if the inverse was computed for an input lying between $\{-1, +1\}$ and the nonlinear polynomial in that region is inverted to yield an overall linear solution, the transfer function for regions below -1 and above $+1$ are even more nonlinear than before because of the existence of higher-order terms. In real applications, if an input signal leaks outside $\{-1, +1\}$ the actual nonlinear signal that is created can be quite high.

Lastly, the noise properties of a nonlinear distortion function can be quite pernicious compared with the noise present in systems with only linear distortion. Because linear distortion is composed of terms that have signal powers of 1, any Gaussian noise present remains Gaussian, while noise power could in many cases get gained up because of the distortion. In nonlinear functions, when a signal-plus-noise term is raised to a higher power, cross-products of the noise and signal are generated that render the noise multiplicative instead of just additive. Other terms are generated that have only higher-order powers of the noise that cause non-Gaussian noise to be present at the output, which severely increases bit-error rates in communications systems. When a *p*th-order inverse is computed and used to lin-

Plot of first five Chebyshev polynomials

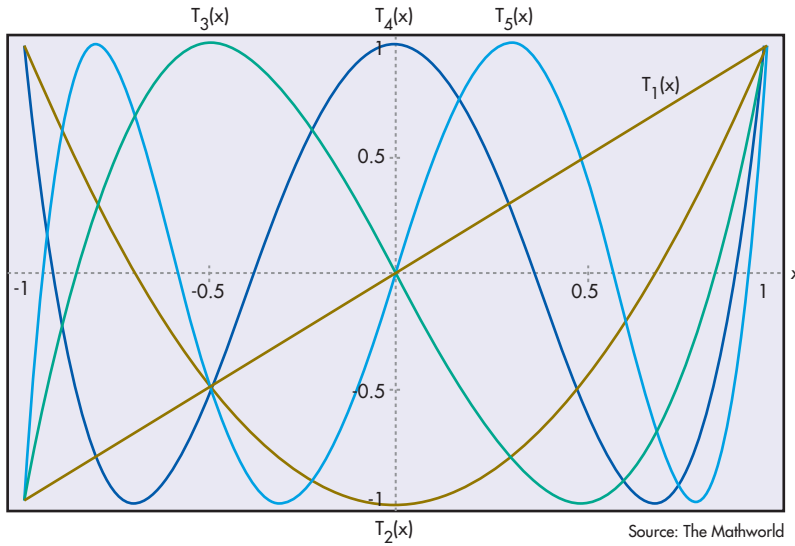


Figure 11

The first four Hermite polynomials

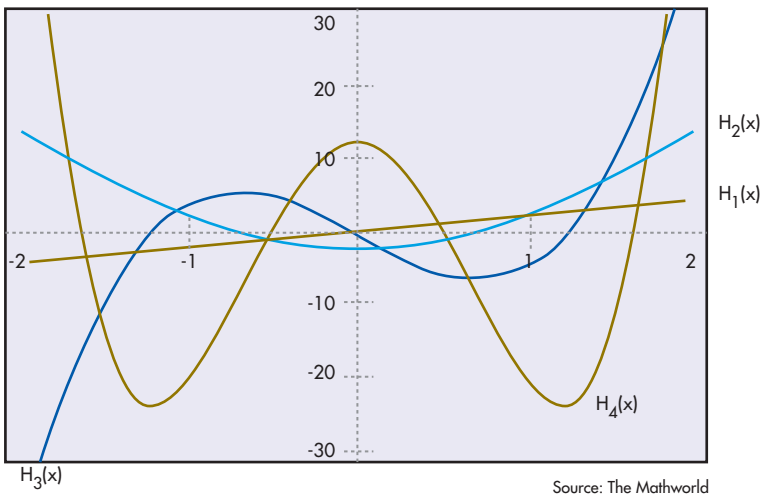


Figure 12

erize the system, the terms that are higher order than p add a significant boost to the noise power and to the deviation from Gaussian statistics.

All of these problems and more (such as the computational requirements for an estimator or inverter due to the higher-power terms in a nonlinear function) illustrate the range of challenges you'll encounter when dealing with even the simplest forms of nonlinear distortion.

MEMORYLESS DISTORTION WITH ORTHOGONAL POLYNOMIAL REPRESENTATION

If the nonlinear distortion function is of the polynomial form, there's an alternative

representation that has certain advantages. It uses *orthogonal polynomial decomposition* of the nonlinear function. The best orthogonal polynomial for a given model depends on the type of input signal, not just the distortion function. For sinusoidal inputs Chebyshev polynomials are the natural choice and for Gaussian signals Hermite polynomials form the basis of the orthogonal expansion.

The expression for the nonlinear model has the form:

$$y_n = x_n + \sum_{j=1}^N k_j T_j(x_n) \quad (10)$$

where $T_j(x_n)$ is the j th Chebyshev polynomial, and k_j is its coefficient. The first few Chebyshev polynomials are:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \end{aligned} \quad (11)$$

Figure 11 shows a plot of the first five Chebyshev polynomials.

The nonlinear function that uses Hermite polynomials has the same structure as the one that uses Chebyshev polynomials, and is given by:

$$y_n = x_n + \sum_{j=1}^N k_j H_j(x_n) \quad (12)$$

The first few Hermite polynomials are:

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_2(x) &= 4x^2 - 2 \\ H_3(x) &= 8x^3 - 12x \end{aligned} \quad (13)$$

Figure 12 shows the plot of the first few Hermite polynomials.

There are a few advantages to the orthogonal polynomial representation. One is that a least squares solution to a channel estimation problem is less prone to local minima. Another is that a p th-order inverse results in the complete nullification of the distortion components below the order P . This is not a property of the generalized polynomial method.

The orthogonal polynomial expansion suffers from the serious disadvantage that a set of polynomials can be designed only for a target signal and must be altered if the signal changes in any manner (such as amplitude or probability distribution). It gets worse. Other negatives are:

- More complicated polynomial forms are needed when the input signal is neither sinusoidal nor Gaussian
- The accuracy of the expansion when using one type of polynomial is sensi-

Equation 14 The Volterra Filter Structure

$$y_n = x_n + \sum_{j_1=0}^N k_1(j_1) x_{n-j_1} + \sum_{j_1=0}^N \sum_{j_2=0}^N k_2(j_1, j_2) x_{n-j_1} x_{n-j_2} + \sum_{j_1=0}^N \sum_{j_2=0}^N \sum_{j_3=0}^N k_3(j_1, j_2, j_3) x_{n-j_1} x_{n-j_2} x_{n-j_3}$$

Equation 15 Third term in the Wiener Filter Structure

$$\sum_{j_1=0}^N \sum_{j_2=0}^N \sum_{j_3=0}^N k_3(j_1, j_2, j_3) x_{n-j_1} x_{n-j_2} x_{n-j_3} - 3\sigma^2 \sum_{j_1=0}^N k_3(j_1, j_1, j_1) x_{n-j_1} - \sigma^2 \sum_{j_1=0}^N \sum_{\substack{j_2=0 \\ j_2 \neq j_1}}^N k_3(j_1, j_2, j_2) x_{n-j_1}$$

tive to any deviation between the input signal statistics and those of a sinusoidal or Gaussian model

- It still has all the disadvantages of noise sensitivity, complexity, and limited range that the general method has.

NONLINEAR DISTORTION WITH MEMORY: THE VOLTERRA STRUCTURE

The Volterra series expansion of nonlinear distortion function is a powerful and generalized structure that covers a very broad class of practical distortion models. Conceptually it's nothing more than a linear system model summed with other "linear" transfer functions whose inputs are higher-order terms of the input signal. Mathematically it's expressed as (in the sampled-data domain) shown in Equation 14.

This series expansion of a nonlinear model clearly covers present and past time signal samples and therefore accounts for nonlinear distortion that has memory effects. It's a superset of the memoryless polynomial model. The nonlinear terms in the Volterra expansion have not only signal terms raised to powers 2, 3, 4, and so on, but also cross-products such as $x_{n-j_1} x_{n-j_2}$ and triple products, and so on.

More elaborate Volterra models involve simplifications of the expansion such that the m th order coefficient is a product of all the previous coefficients. Alternatively, the m th order coefficient is a power of m of the first coefficient. Either way, the Volterra series can be viewed as linear distortion with memory cascaded with a memoryless nonlinear distortion.

The advantage of the Volterra model is

its ability to represent very severe practical nonlinear distortion functions with memory. Disadvantages include:

- Its complexity
- Its very high computational processing requirement
- The difficulty of fitting a model into a measured nonlinear channel because of the cross-coupling of signal terms
- The even greater difficulty of fitting a model into measured data if the number of terms chosen for the Volterra equation results in an under- or over-modeled channel
- The difficulty of adapting a filter to a model because the cross-coupling of terms results in local minima in the error surface and therefore a suboptimal performance
- The inversion problem similar to the polynomial fit one where a p th-order inverse still leaves higher-order terms, and even creates higher-order terms that were not present before the inversion of a nonlinear channel.

NONLINEAR DISTORTION WITH MEMORY: THE WIENER STRUCTURE

The Wiener structure attempts to improve the ability to fit a series expansion into a nonlinear model by transforming the Volterra series into an orthogonal equivalent series. The Wiener expansion is similar to the Volterra series except that each term has the effects of the lower-order terms subtracted from it in order to make it independent. Since all terms are independent the series is rendered orthogonal. For exam-

ple the third term in a Wiener series is shown in Equation 15.

Other than the advantage of having better properties when fitting it to a nonlinear channel, the Wiener model suffers from all the same disadvantages as the Volterra model and is more complex mathematically and therefore requires ever more processing power.

ROOM FOR IMPROVEMENT

Nonlinear distortion models are complex and suffer from many difficulties that render them impractical to apply to real engineering situations. Channels that suffer from nonlinear distortion can be modeled using higher-order polynomials or series. The models discussed here attempt to perform system identification or inversion by using those higher-order expansions as parallel models or inverse models to the channels at hand. All of them demand a lot of processing power to compute higher-power signal terms and because of the large number of terms in the series expansion. The cross-coupling between terms means that a precise fit between model and channel is often fraught with inaccuracy that diminishes the potential for adaptive forms of nonlinear filters. ■

Roy Batruni is founder and CEO of Optichron, which specializes in the problem of nonlinear distortion. Batruni has 25 years experience in the field. he has an MS in electrical engineering from Cornell University and an MBA from the Haas School of Business, UC Berkeley.

**OPTICHRON**