

Signals and Measurements for Disk Drive Design

Table of Contents

[Preface](#)

[Introduction](#)

[1. The Read Channel Signal](#)

[2. Creating the Transition Response](#)

[3. Creating the Data Pattern](#)

[4. Merging the Transition Response with the Data Pattern](#)

[5. Appending Other Data to the Waveform](#)

[6. Inserting Amplitude Variations](#)

Creating Missing/Extra Bits

Building a 1M-Point Waveform Sequence

Simulating Mechanical Shock

Simulating Residual 1F Amplitude After 2F Overwrite

Using a Filter to Modify Amplitude Characteristics

Using an FFT to Modify Amplitude Characteristics

[7. Inserting Noise Impairments](#)

Using the AWG Noise Waveform

Verifying Read Channel Frequency Response

Simulating Signal-to-Noise Ratios

Making Coherent Noise More "Random"

[8. Inserting Timing Impairments](#)

Simulating Peak Shift (Jitter)

Simulating Frequency Shift to Verify Clock Recovery Loops

"Beating the Spec" to Achieve Higher Resolution

[9. Appendices](#)

Related Tektronix Publications

Addendum--Region Shift

Tektronix Product Features & Specifications

Tektronix Measurement products are manufactured in ISO registered facilities.

Preface

As the performance criteria for each successive generation of hard disk drives become more stringent, the test equipment used in disk drive design must keep pace. This applications booklet describes the use of two essential tools--the arbitrary waveform generator and the digitizing oscilloscope--in solving some of the most demanding disk test problems encountered today.

Arbitrary waveform generators (AWGs), with their ability to create almost any waveform imaginable, have become an indispensable design and test tool in the engineer's kit. The Tektronix 2000 Series AWGs are especially well-equipped to deal with the unique problems encountered in disk drive testing. In particular, the AWG 2041 provides the clock rate and record length needed for the most advanced rotating media testing requirements.

The 2000 Series products offer much more than just raw performance. Each of the three AWG models in the series--the AWG 2041, the AWG 2021, and the AWG 2005--has a powerful graphical user interface built in. Their screen-based control panels are coupled to soft keys that access such functions as range and mode selection, display format, and marker placement. Scrolling, cut/paste selection, cursor position, and file selection are controlled by a continuous knob on the instrument's front panel. Every pertinent function, status, and value is available on the screen, only a menu away.

Another important facet of the 2000 Series is its wide range of waveform editing tools. Most conventional AWGs require an external computer and software for complex waveform editing. Others provide only simple front panel tools, like keypad formula entry. In contrast, the 2000 Series includes seven built-in ways to edit, as well as compatibility with Tek's powerful WaveWriter software for external computers. Several of these integrated waveform tools are especially useful for disk drive testing: the graphical editor, the FFT editor, the equation editor (ideal for producing Lorentzian pulse waveforms), and the sequence editor. Each of these functions is discussed in detail in this booklet. Also of interest is the 2000 Series direct DSO transfer capability, which permits waveforms acquired from a reference source (such as a known-good UUT) to be loaded into the AWG and reproduced.

The TDS 744A digitizing oscilloscope is the perfect complement for the AWG 2041 in disk drive applications. The TDS 744A's record length of up to 500k is eminently suited for storing multiple waveform cycles. The scope's 2 GS/s sample rate and peak detect capability help isolate the tiny glitches and view fast signal edges. And of course, the TDS 744A offers direct waveform transfer compatibility with the 2000 Series AWGs.

The test problems described here are "real world" examples that reflect today's concerns among designers and manufacturers of high-speed, high-capacity hard disk drives. The solutions, though, are also useful for a host of design verification applications for products ranging from cellular telephones to biomedical monitors.

Introduction

To remain competitive, disk drive manufacturers must continue to increase bit densities and transfer rates. Designers need to carefully evaluate the complex interactions among improvements in media, head, and data recovery electronics. One of the most challenging tasks during drive electronics development is the generation of realistic read channel signals to simulate nominal and worst case conditions. Traditional swept sinusoid or pulse response analysis is simply inadequate. Because real-world disk drive signals are quite complex, simplified models typically produce incomplete results. Test equipment manufacturers have responded to this application with the arbitrary waveform generator (AWG). The AWG provides drive manufacturers with the flexibility to define and generate test waveforms with almost infinitely variable signal characteristics. The AWG is the signal generator equivalent to the computer spreadsheet: the user can create countless "what-if" waveforms to thoroughly evaluate or test new concepts, prototype circuits, or production assemblies. Ironically, the inherent flexibility of an AWG can make it difficult to select a model that will fit a specific application. For example, you will not find a specification that explicitly defines an AWG's ability to generate a particular servo pattern. In general, an AWG's ability to generate a specific signal must be demonstrated by example.

The most straightforward method to create a specific signal is to use the record/playback technique. A live signal is recorded into the memory of a digital oscilloscope. The record is transferred to the AWG for playback. This method is expedient but has limited flexibility. Creating and editing a customized signal is a more powerful technique and is the focus of this note. The first step is to *create* the basic magnetic channel signal. The next step is to *edit* the basic signal by modifying amplitude and timing parameters to simulate non-linearities and aberrations. In this note we use the AWG 2041 because it provides the signal generation capabilities most suitable for present and evolving drive application requirements. The AWG 2041 also has the built-in creation and editing functions relevant to disk drive applications. For waveform capture and analysis, we use the Tektronix TDS 744A digital oscilloscope, which complements the capabilities of the AWG 2041.

1. The Read Channel Signal

Traditional function generators only satisfy the basic characterization requirements of read channel circuits. The real-world signals presented to the read channel circuits are much more sophisticated and contain aberrations that are virtually impossible to create with traditional instruments. But, using the basic magnetic channel waveform as a building block, it is remarkably easy to develop a wide variety of realistic test signals.

Virtually all magnetic disk drives use saturated recording, where each "bit" interval on the media is polarized in one of two states. Information is implicit in the *transitions* between states. That is, data is encoded by simply changing the sign of the current in the write head, which flips the polarity of the magnetic media below it. Later, the read head senses the flux reversals on the media and generates a response to the transition. A read

channel signal can be modeled as a string of transition responses. Thus, the first fundamental component in building the read channel waveform is the transition response.

The second component is the digital data pattern, which defines the locations of the transitions. This pattern can range from a simple 1F or 2F single-frequency pattern to a $2^n - 1$ maximal length pseudo-random pattern. The two components are then combined to create the channel waveform.

Eq. 1

$$v(t) = \frac{1}{1 + [2tx]^2}$$

Eq. 2

$$v(t) = \frac{1}{1 + \left(\frac{2(t-t_0)/T}{PW50/T} \right)^2}$$

A popular and effective model of the transition response in longitudinal recording is the Lorentzian pulse. This is the voltage response at the output of the read head in response to an isolated transition on the media. The sign of the pulse alternates for successive flux reversals. Equation 1 shows the basic Lorentzian function, where t is time and x is a parameter which controls the pulse width. The pulse has a maximum value at $t=0$ and symmetrically decays to zero about $t=0$. Equation 2 is a slight modification that allows direct entry into the AWG 2041's equation editor. The parameter t_0 shifts the pulse so that it is non-zero for only positive time. The pulse peak now occurs at $t=t_0$. $PW50$ represents the pulse width where the amplitude is 50% of the peak level. The "dummy" parameter, T , inserted in the numerator and denominator of the squared term represents the bit interval. This insertion permits the system design ratio, $PW50/T$, to appear as an explicit term in the equation.

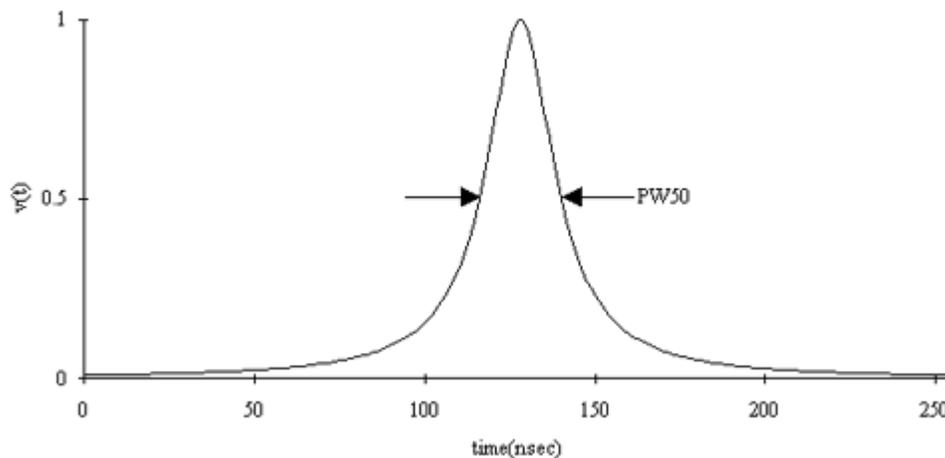


Figure 1. The Lorentzian pulse with $T=20$ ns and $PW50/T=1.2$. This is our model of the read-head voltage response to a write-current transition.

Figure 1 plots the Lorentzian function for a bit interval of $T=20$ ns and a $PW50/T$ ratio of 1.2. Larger values of $PW50/T$ spread out the pulse. A total duration of 256 ns adequately represents the transition pulse. The pulse was centered at $t_0=128$ ns. Note that the response to an instantaneous write-current transition extends beyond the bit interval in which the transition occurred.

2. Creating the Transition Response

Figure 2 defines this pulse in the AWG 2041's equation editor. The first line defines the pulse to be 256 ns long. The next two lines introduce the two constants k_0 and k_1 to represent T and $PW50/T$. This provides a simple way to change these parameters later. The fourth line duplicates Equation 2 with the t_0 timing offset of 128 ns.

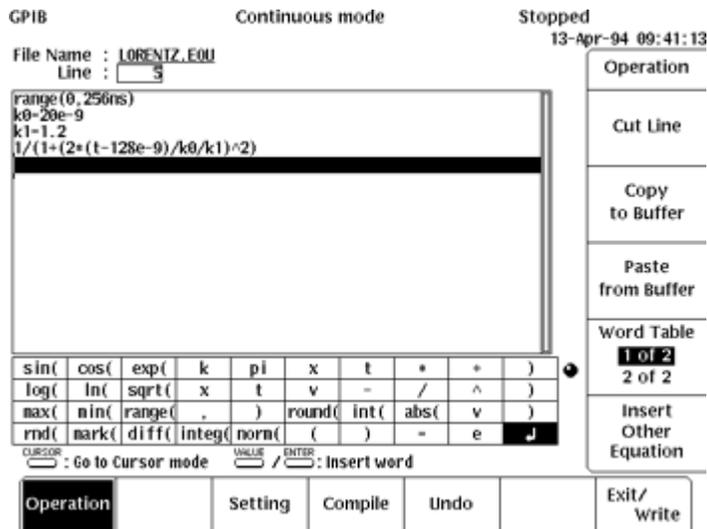


Figure 2. The pulse is defined in the AWG's equation editor.

The AWG's compiler converts the equation into a series of points and displays the result (Figure 3). In this case, the 256 ns pulse duration is converted to a 256 point record. We could generate this pulse as an AWG output signal, but the transition pulse is only a building component. The next step is to define the data patterns which created modulated waveforms. Note that we are modeling the *transition response for longitudinal recording*. We could also create a transition response for perpendicular recording systems, which has the approximate shape of the derivative of the Lorentzian pulse--in that a write transition yields a positive-negative pulse pair.

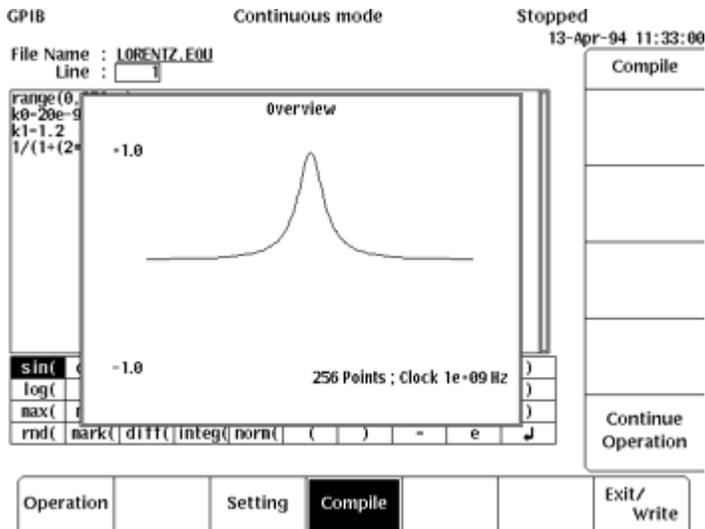


Figure 3. The AWG 2041 equation compiler calculates and plots the series of 256 points corresponding to the selected pulse parameters. Since the AWG clock rate was set to 1 GS/s, each point is separated by 1 ns.

3. Creating the Data Pattern

The most practical model of the data pattern emulates the write-current, since this represents a physical signal in the write-read process. The write-current binary pattern has the fundamental period of the bit interval T and is the final pattern *after* any coding algorithms (e.g., 2,7 RLL). For our example we will use a 62-bit write-current pattern represented by the sequence, "1101100... (48 other bits) ...1010011." You can directly enter this NRZ binary pattern using the AWG 2041's pattern editor (Figure 4). If you have the binary pattern *before* coding, then you can use the AWG 2041 editor's built-in code converter, which provides common translations such as MFM and 2,7 RLL. You can also define your own conversion table to implement other translations.

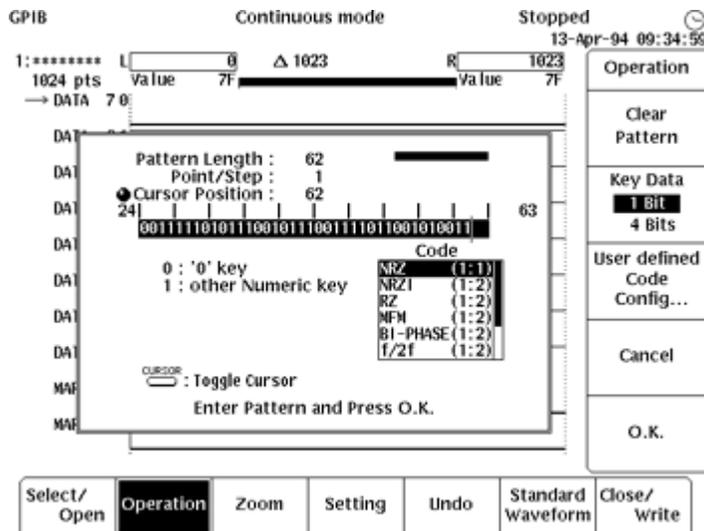


Figure 4. The pattern editor enables direct entry of binary data patterns.

Once you have the write-current pattern, you need to scale the signal to match the target bit-interval. For a bit-interval of $T=20$ ns, each pattern bit represents 20 ns. If we use the AWG's 1 GS/s clock rate, with which each data point represents 1 ns, then each pattern bit must be expanded to occupy 20 data points in the AWG memory. You use the horizontal scaling function in the AWG to expand the 62-bit pattern into a 1240-point record with each original point simply repeated 20 times. The final 1240-point write-current pattern can be displayed graphically, or we could generate this pattern as an AWG output signal (Figure 5).

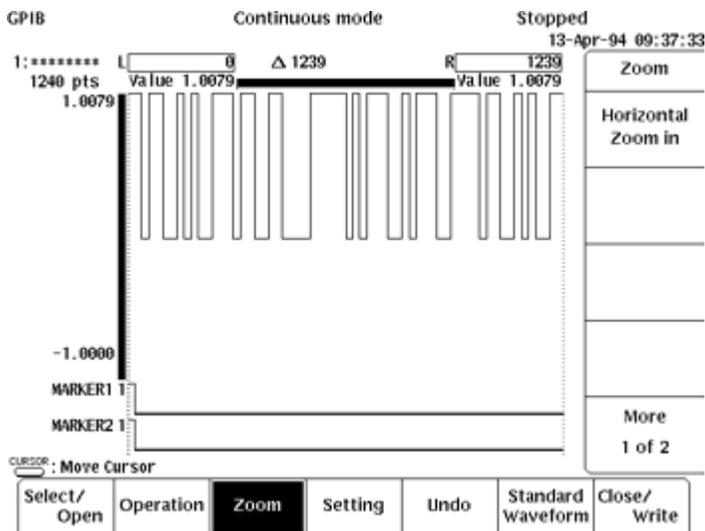


Figure 5. Graphical view of the 62-bit write-current data pattern. Each bit interval is 20 ns long. If the AWG clock rate is set to 1 ns per point, each data bit occupies 20 memory points. The physical write-current waveform is actually a symmetrical bipolar signal, but we are interested only in the locations of the transitions.

4. Merging the Transition Response with the Data Pattern

The final step is to combine the transition response with the write-current pattern (Figure 6). We will use the linear superposition assumption to combine the pulse responses with individual write-current transitions. In other words, we will convolve the write-current transition waveform with the transition response waveform. The AWG 2041's convolution function directly performs this function with an important option. Since the write-current pattern represents the physical write-current, we need to convert this to a pattern of write-current transitions. Thus, we apply the differential option of the convolution. This means that a positive transition pulse is summed into the final waveform on a 01 pattern transition and a negative transition pulse is summed on a 10 pattern transition. A continuous string of 1's or 0's produces no response.

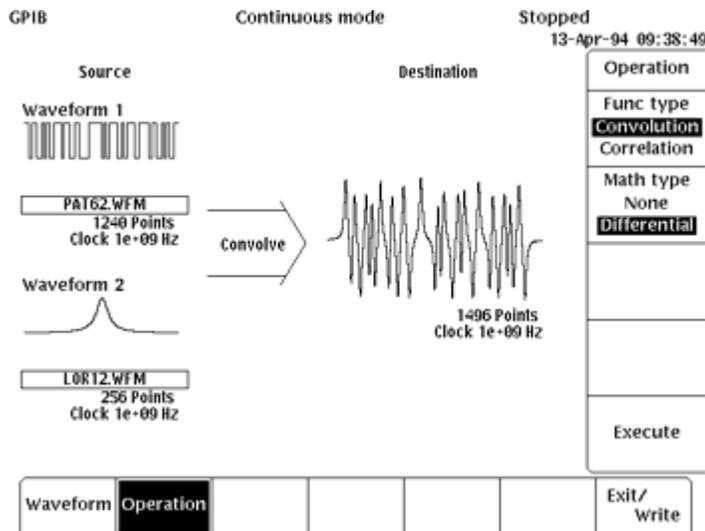


Figure 6. The AWG's differential convolution function directly calculates the effect of each write-current transition.

The convolution of the two waveforms produces a new waveform that is 1496 points long (Figure 7). This is the sum of the two individual waveform lengths and is a by-product of the convolution process. Note that a write-current transition occurs over a 1-bit interval, while the result of the transition occurs over 256 intervals. Thus, the effect of a transition near the ends of the write-current pattern extends beyond the original pattern interval. This highlights an often confusing issue when creating an arbitrary read channel waveform. We started with an isolated 62-bit pattern occupying 1240 ns and ended up with a 62-bit pattern occupying 1496 ns! How would you *repetitively* generate the

original 62-bit pattern in the allotted 1240 ns? Should you simply truncate the 1496 ns record to a 1240-point record and concatenate the shorter record?

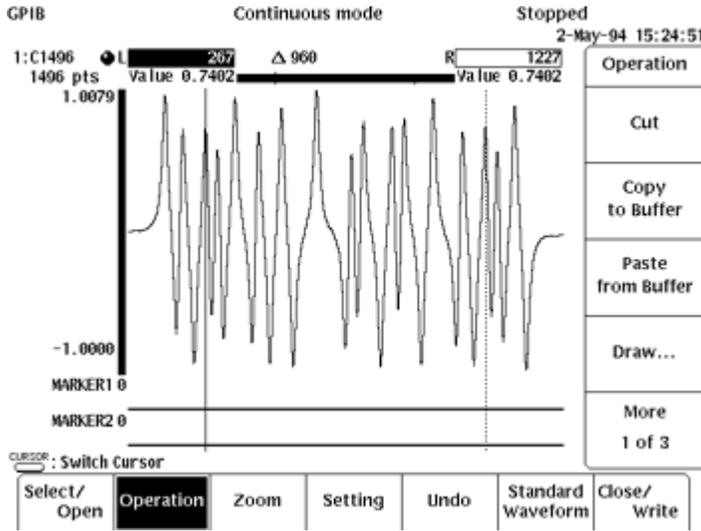
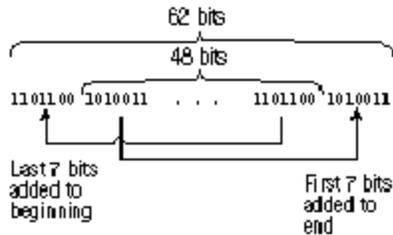


Figure 7. In differential convolution, each write-current transition adds the 256 ns transition pulse response to the final waveform. Each pulse is centered at the location of the write-current transition that caused it. The sign of the added pulse depends on whether the transition was from 0 to 1 or vice versa.

As modeled by the transition response pulse, an instantaneous write-current transition affects a time interval that is several times longer than the fundamental bit interval. Our model requires knowledge of write-current transitions immediately before and after a finite length test pattern. Returning to the example, it turns out that our basic write-current pattern was actually 48 bits long.



The basic 48-bit pattern was padded with 14 bits (7 bits at the beginning and 7 at the end) to create the final 62-bit pattern. The 7 bits at the beginning of this 62-bit pattern are duplicates of the last 7 bits in the 48-bit pattern, while the 7 bits at the end are duplicates of the first 7 in the 48-bit pattern. The 7-bit overlap was chosen because it represents 140 ns (i.e., 20 ns per bit interval). The total transition pulse time was modeled as 256 ns, so the effect of a write-current transition extends 128 ns before and after the transition.

Thus, to generate the 48-bit test pattern repetitively, we have to pad both sides of the pattern with any data transitions whose transition response could fall into the 48-bit interval. In this case, 7 bit intervals, or 140 ns, on each side adequately covers the 128 ns tail of the transition pulse. Thus, our final waveform is really 48 bit intervals of 20 ns each, or 960 ns long. Using the AWG's graphical editor, we truncate the 268 points equally off each end of the 1496-point convolution result. Note that the points at the ends of the truncated 960-point segment form a seamless boundary. That is, if we repetitively generate this particular pattern, the AWG output will be smooth and continuous between successive 48-bit patterns.

We now have the tools to quickly create building block segments of read channel waveforms. For example, by simply changing the $PW50/T$ parameter in the Lorentzian pulse equation, we can generate the read channel waveform for the same write-current pattern but with a different head-media interface (Figure 8). Conventional waveforms, such as 2F or 1F patterns, can be created by using a write-current pattern of ...10101010... or ...11001100... to simulate transitions on each or every other bit interval. This transition response model of the write-read process does not preclude the explicit entry of an equation model. For example, all AWG equation editors will create cosine-cubed or sinusoidal bursts to simulate unmodulated test signals or servo patterns.

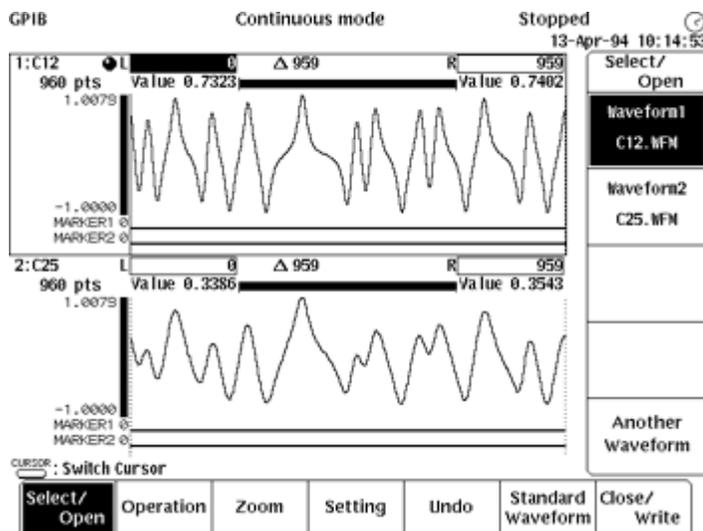


Figure 8. By changing the $PW50/T$ parameter in the Lorentzian equation to 2.5, a new response is quickly developed. The AWG's editor simultaneously displays the 960-point read channel waveforms for $PW50/T=1.2$ (upper) and $PW50/T=2.5$ (lower). As expected, individual pulse response peaks are less well defined. In this case, the waveform makes an ideal test signal for partial response detection. The same 62-bit padded data pattern was used so that the final 48-bit, 960 ns waveform is also continuous at the boundaries.

5. Appending Other Data to the Waveform

Our 48-bit example introduced the data continuity issue, which frequently confuses first-time AWG users. A conventional signal generator creates seamless repetitions of basic waveshapes, but modulated read channel waveforms are not basic waveshapes! If the application is the repetitive generation of a phase continuous data pattern, then you need to plan your waveform editing with this in mind. If you already have several arbitrary waveforms that you need to concatenate, you might be lucky enough to find that they seamlessly connect. But you will probably get better results by stringing together the original data patterns and then convolving the longer pattern with the transition pulse response. A typical application might be to string together a preamble pattern, address and other sync patterns, and sector data patterns *before* the convolution step.

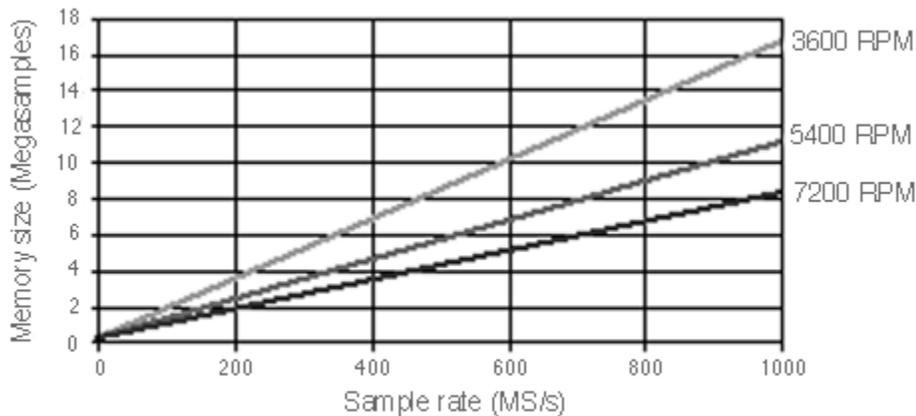


Figure 9. Memory requirements to simulate an entire track of data can be overwhelming!

Keep in mind that irrespective of the method of creating a waveform, AWG memory size will ultimately limit the length or size of an arbitrary waveform. The desire to completely simulate the waveform from a full disk drive revolution must be carefully evaluated. For example, running at the 1 GS/s sample rate on a 5400 RPM drive, one revolution requires a memory size of about 11 million points (11 MS), exceeding the capacity of any AWG currently available. The AWG 2041 provides only up to 4 MS, so you could simulate less than half a revolution. On the other hand, if you dropped back to a 300 MS/s sample rate, you could simulate the entire track (Figure 9).

6. Inserting Amplitude Variations

We will first demonstrate techniques to insert amplitude or level variations into the basic channel waveform. A practical approach for developing aberration models is to categorize them by duration. For example, a media defect might affect only a few bit intervals, while flying height modulation might last over an entire revolution. By starting with a basic read channel waveform, whether data modulated or a 1F waveform, our basic approach for inserting errors is to first select a time segment or slice of the overall

waveform. Then we simply multiply the original waveform by the aberration or add the aberration to the waveform.

Creating Missing/Extra Bits

The case of the missing or extra bits is straightforward. You can vertically scale the amplitude of a waveform segment using the AWG editor (Figure 10). In this example, the cursors select a 110 ns slice and the vertical scaling function decreases the amplitude of the entire segment by 50%. The AWG 2041's marker outputs are also shown in this example. The bits of two markers are associated with each waveform point. These provide two binary output signals that can assist in triggering instrumentation. Marker 1 was set to generate a pulse at each 20 ns bit interval, and Marker 2 was set to generate a pulse during the amplitude aberration. Marker 1 could be used to synchronize a scope to view the data recovery clock jitter, and Marker 2 can trigger the capture of a downstream signal in the read channel circuit.

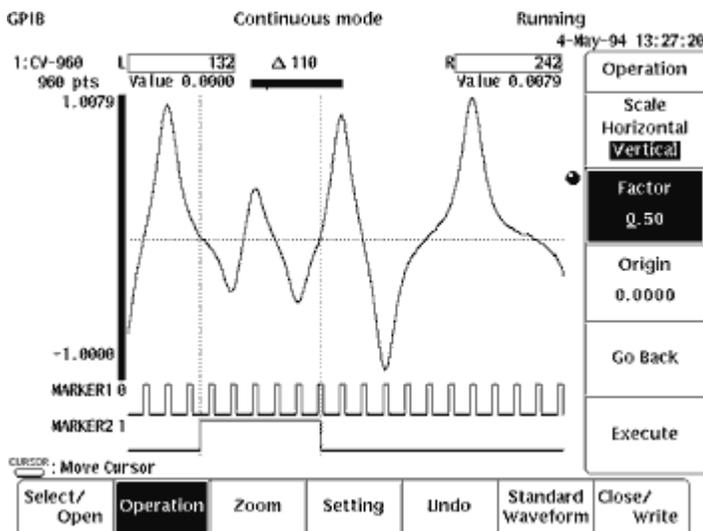


Figure 10. The vertical cursors define the duration of the aberration. All points in the segment are scaled by 0.5 to simulate missing bits. The left (L) and right (R) cursor readouts show that the affected area falls between the 132nd and 242nd points in the 960-point waveform.

Building a 1M-Point Waveform Sequence

Since media defects are relatively infrequent, we use the AWG's sequence editor to generate this aberration with a low duty cycle. In this case we have two waveforms in our repertoire. The first is our 960-point (48-bit) test pattern waveform, CV-960.WFM, and the second is the same waveform plus the "missing" bits, CV-960M.WFM. The sequence editor creates a script for combining individual segments (Figure 11). In this example, we define a sequence where the basic test waveform repeats 1000 times, the version with the

missing bits is inserted once, then the sequence repeats. Note that this simple script defines a new waveform with an equivalent size of 960,960 points!

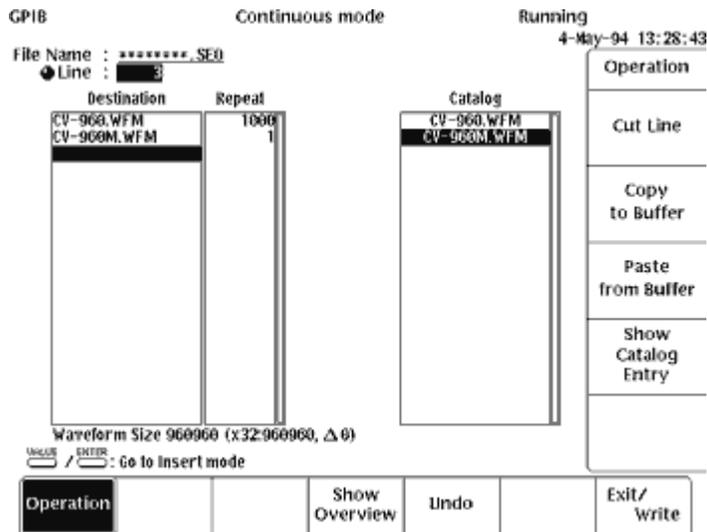


Figure 11. The AWG's sequence editor provides a simple scripting procedure for concatenating waveform blocks.

Simulating Mechanical Shock

We need the sequence editor to generate long waveforms for applications that simulate mechanically induced amplitude modulations. Suppose we want to simulate a mechanical shock in a small laptop drive. We will use a damped sinusoid to model the shock's effect on the signal amplitude (Figure 12). Four lines in the equation editor implement the model. First, we define a 1 ms waveform using the range command. Then we define two constants that will simplify changes to the basic ring and decay parameters. The ring frequency is set to 5000 Hz using $k0$. The exponential decay constant is set to 200 μ s using $k1$. The fourth line defines the damped sinusoid using these two constants. The DC term of 0.5 creates a positive-only multiplier so that the nominal level is non-zero.

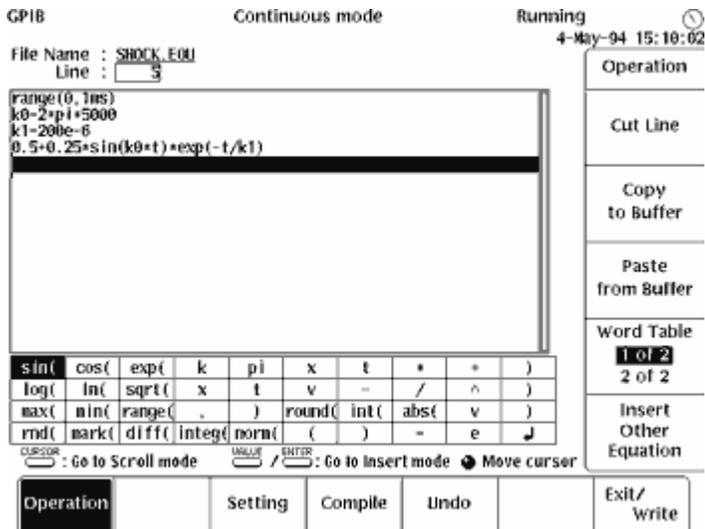


Figure 12. An equation editor defines a damped sinusoid to amplitude modulate the read channel waveform.

Since the damped sinusoid will amplitude modulate the data waveform, we need to match the lengths of both waveforms. In this example we will use an AWG clock rate of 200 MS/s or 5 ns/point. The sequence editor was used to create an expanded data waveform consisting of 200 repetitions of the basic 960-point waveform. This is 192,000 points or just about 1 ms. We use the waveform editor to expand the damped sinusoid to the same record length of 192,000 points. The exact length of the two waveforms is 960 μ s. Then we simply multiply the two waveforms together to perform the modulation (Figure 13).

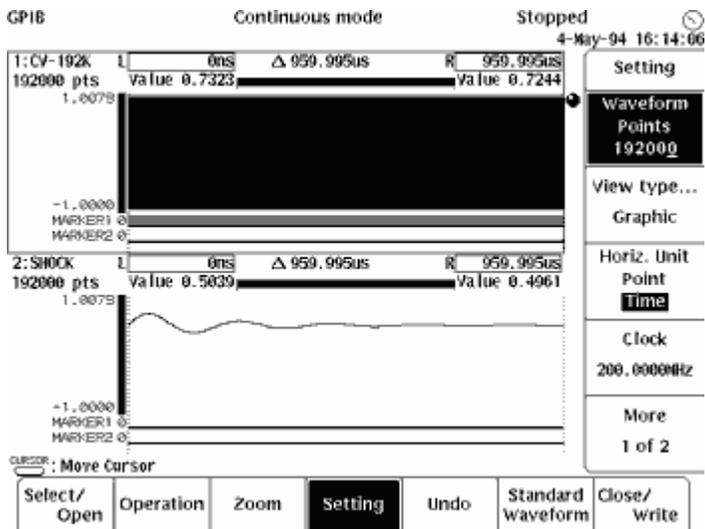


Figure 13. The AWG's editor displays the two waveforms that will be multiplied together. The record lengths of the two waveforms must match.

A marker pulse was added to the final waveform at the beginning of the modulation envelope to simplify scope triggering. The TDS' peak detect acquisition readily captures the envelope (Figure 14). The cursors verify the 5000 Hz ring frequency.

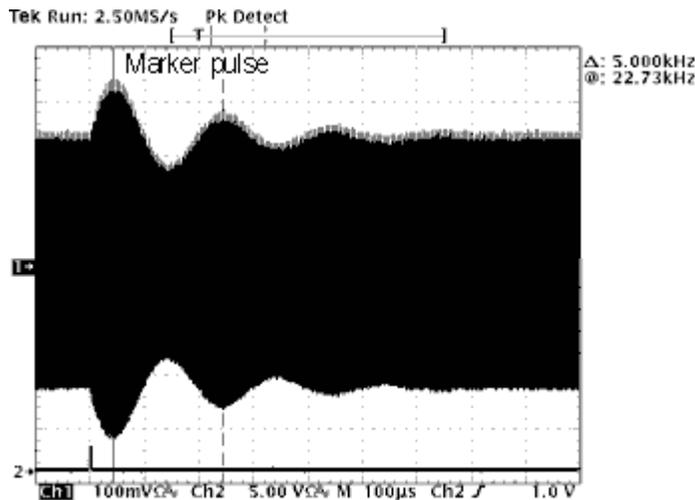


Figure 14. The modulated read channel signal as captured by the TDS 744A oscilloscope.

Simulating Residual 1F Amplitude After 2F Overwrite

The next example simulates the residual 1F amplitude after a 2F overwrite. Since a typical 1F residual is attenuated by at least 30 dB, the approach is to add an attenuated 1F waveform to a 2F waveform. The editing procedure is similar to the previous example except we are adding instead of multiplying two waveforms (Figure 15). First, we attenuate the 1F waveform by using the editor's vertical scaling function. Then we simply add the two waveforms. The basic 1F and 2F patterns were created using the convolution procedure described above. As before, the record lengths must match in order to add the waveforms point-by-point. In this case, we could have used a common 80-point record, since the slower 1F pattern repeats in this interval. However, we used a 960-point common record length for consistency with our modulated data waveform.

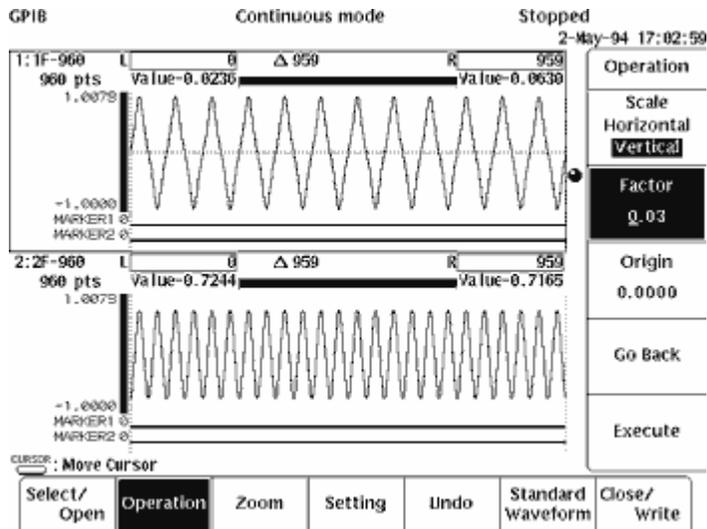


Figure 15. The 1F waveform (upper) is scaled by 0.03 (30.5 dB) before it is added to the 2F waveform (lower).

The TDS 744A display of the AWG overwrite signal is shown in Figure 16. The upper trace is the original 1F waveform. The time scale is 100 ns/div. The second trace is the FFT of the 1F waveform. The TDS' gated measurement function zooms in on the fundamental and measures a level of 12 dBV centered at 12.5 MHz as expected. The absolute level is unimportant since we are interested in only the attenuation in the overwrite waveform. Note that the vertical scale of the FFT results are 40 dB/div and the horizontal scale is 10 MHz/div. The third and fourth traces are the overwrite waveform and its FFT. The 12.5 MHz fundamental is now 44.8 dBV, so it has been attenuated by 32.8 dB. While the attenuation is not exactly 30 dB, the results are within allowable tolerances and are quite good given the finite 8-bit resolution of the 1 GS/s AWG generator. Note that the computational errors due to finite resolution occur only once. The 1F/2F relationships are not constant in the digital waveform record.

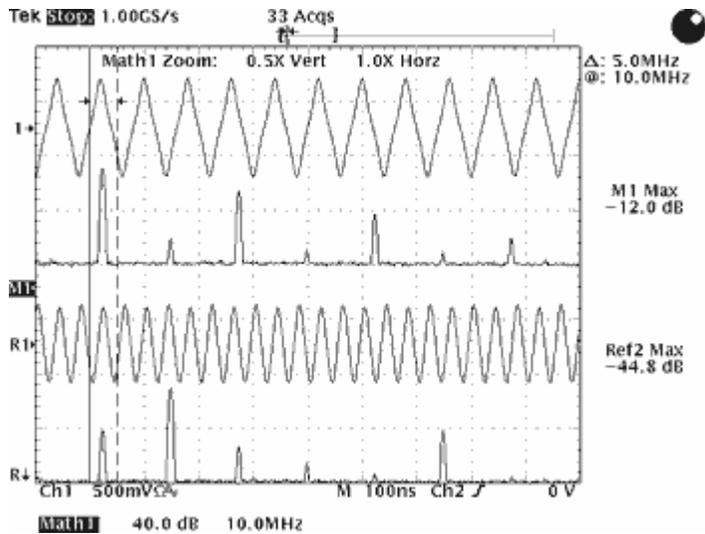


Figure 16. TDS 744A captured the original 1F and the calculated overwrite signal. The FFT calculates the levels of the 1F fundamental at 12.5 MHz.

Using a Filter to Modify Amplitude Characteristics

The previous example introduced the TDS 744A's FFT capability to provide a frequency domain view of an amplitude aberration. In fact, the residual 1F signal aberration was too small to observe directly in a time domain representation! This illustrates the need to modify the amplitude characteristics of test signals through the frequency domain. The AWG 2041 provides two basic methods for altering the amplitudes of the frequency components of a test signal. The first method is the straightforward hardware filter. The AWG provides a set of four Bessel low-pass filters: 10, 20, 50, and 100 MHz. The Bessel filter's linear phase response is particularly useful for preserving the peak characteristics of read channel test signals.

Figure 17 shows the Lorentzian pulse through three of the filters. This is the same 256 ns pulse we used to develop the modulated data waveform. The leftmost peak is the unfiltered pulse directly from the equation compiler. The next pulse was applied to the 100 MHz filter. There is a negligible peak amplitude reduction, and the signal is delayed relative to the unfiltered pulse by about 5 ns. If you synchronize the TDS on the AWG's marker pulse, enabling the AWG's 100 MHz filter will delay a modulated data waveform by 5 ns, in addition to filtering the output. The 50 MHz filter attenuates the peak slightly more and delays the peak by about 10 ns but nevertheless preserves the pulse shape. The 20 MHz filter attenuates the peak significantly and broadens the pulse shape, due to the attenuation of its higher frequency components. This filter is inappropriate for our original read channel pulse.

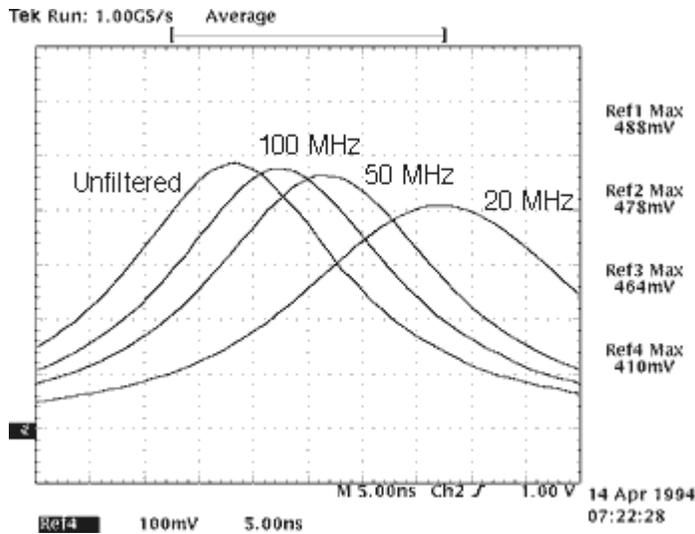


Figure 17. The unfiltered Lorentzian pulse output from AWG is the leftmost peak. The delayed pulses show the effect of the 100 MHz (shortest delay), 50 MHz, and 20 MHz (longest delay) Bessel low-pass filters.

The hardware low-pass filter is easy to apply, but the choices are fixed and may not match your application. In Figure 18, the TDS 744A captures a modulated data pattern with and without the AWG's 50 MHz low-pass output filter. The signal is delayed as predicted, but the attenuation above 50 MHz is barely noticeable in the frequency domain. We have already discovered that the 20 MHz hardware filter is unsuitable, so we need another solution to shape the frequency components in the data pattern.

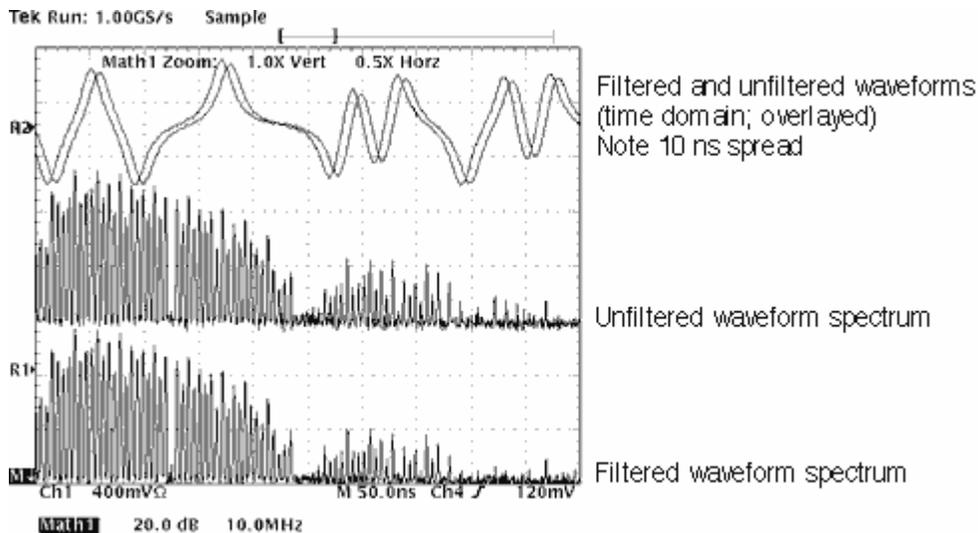


Figure 18. The upper two overlaid waveforms show the modulated data pattern with and without the 50 MHz hardware low-pass filter. Since the marker pulse was used to trigger the TDS, the filtered waveform is delayed by about 10 ns. The TDS calculates the FFT spectra of the two signals in the lower two traces. The bottom trace is the filtered waveform, but the attenuation above 50 MHz is barely noticeable.

Using an FFT to Modify Amplitude Characteristics

The second method of altering frequency components is the AWG 2041's software based FFT editor option. The FFT editor lets you alter the magnitude and phase characteristics of a signal in the frequency domain. Figure 19 shows the AWG's filter editing menu applied to the modulated data pattern. We select the low-pass filter option and need to enter a break frequency and a rolloff rate. Here we enter 50.0 MHz and a brick wall rolloff of 400 dB/octave. The AWG editor uses the FFT to transform the original waveform into the frequency domain. It removes all the components above 50 MHz and transforms the altered frequency domain record back to a time domain record. Note that the editor also provides built-in high-pass, band-pass, and band-elimination filter templates.

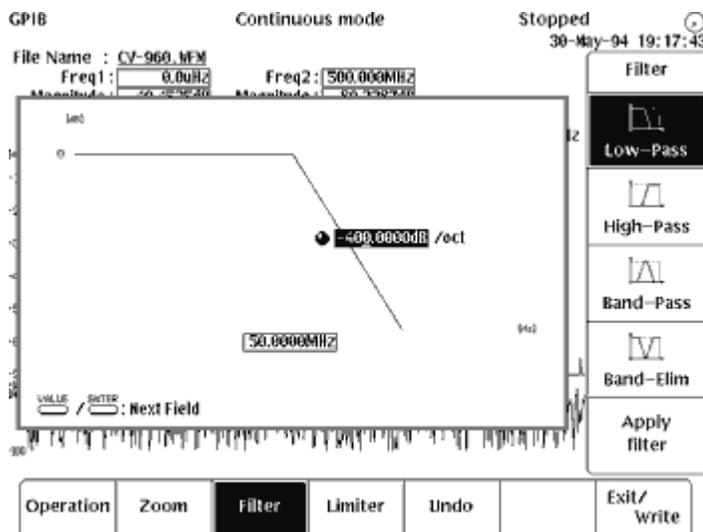


Figure 19. The AWG 2041's FFT editor has built-in filtering functions to shape the frequency domain components of a waveform. Here the low-pass filter function is used to remove frequency components above 50 MHz in the modulated data pattern. The FFT editor is particularly useful in reshaping modulated patterns to verify the performance of channel filters.

The unfiltered waveform and the waveform filtered by the AWG's filter editor are captured by the TDS in Figure 20. The two traces exactly overlay in the time domain records. The two FFT traces reveal the superior filtering performance of the FFT-edited waveform. As in the overwrite example, traditional time-domain scope waveforms do not

reveal that the frequency components above 50 MHz have been successfully removed. Thus, it is the *combination* of the AWG's time-domain and frequency-domain editing tools that provides the power and flexibility to create the signal aberrations required in modern disk drive applications.

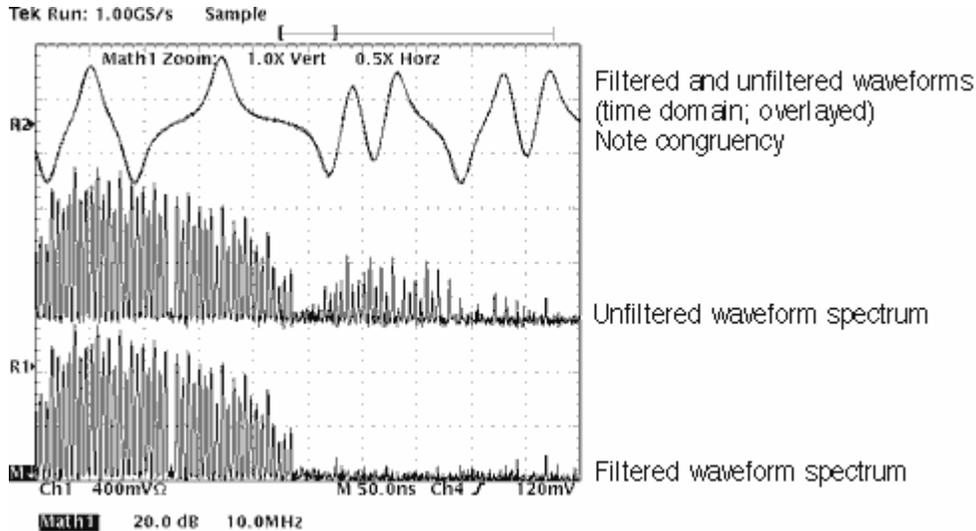


Figure 20. The upper two overlaid waveforms show the modulated data pattern with and without the 50 MHz brick wall filter. The filtered waveform is not delayed relative to the unfiltered waveform. The TDS calculates the FFT spectra of the two signals in the lower two traces. The bottom trace is the spectrum of the filtered waveform.

7. Inserting Noise Impairments

Using the AWG Noise Waveform

Although the removal of noise is a common design goal, a noise source can be an extremely useful test stimulus or signal impairment. The AWG 2041 provides a built-in noise function, but its characteristics are quite different than traditional sources, such as noise diodes. An AWG* noise waveform is actually a calculated series of pseudo-random numbers. There are two key properties of the AWG noise function which we will examine through applications. First, the AWG noise signal is actually a series of voltages that changes once per clock period. This has implications for the spectral characteristics of the signal. The second property arises because an AWG noise waveform is simply another pre-calculated record that must repeat in order to create a continuous signal.

The top trace in Figure 21 shows a pseudo-random noise record of 256K points and a 50 MHz clock. The key clock-related feature is that the waveform appears to be a staircase function. The sharp edges can be removed by the AWG's 20 MHz low-pass filter, as shown in the second trace above. The TDS' FFT spectra for the two signals are overlaid below the two time-domain waveforms. The salient characteristic of the unfiltered noise

spectrum is that it rolls off with a $(\sin x)/x$ function with the first null at the clock frequency and subsequent nulls at multiples of the clock rate. Even below the first null, the noise density is constant only well below the clock frequency. In this example, we used the AWG's 20 MHz hardware filter to attenuate the frequency components above the first spectral null to achieve a bandwidth limited noise stimulus with relatively constant noise density from DC to a few MHz.

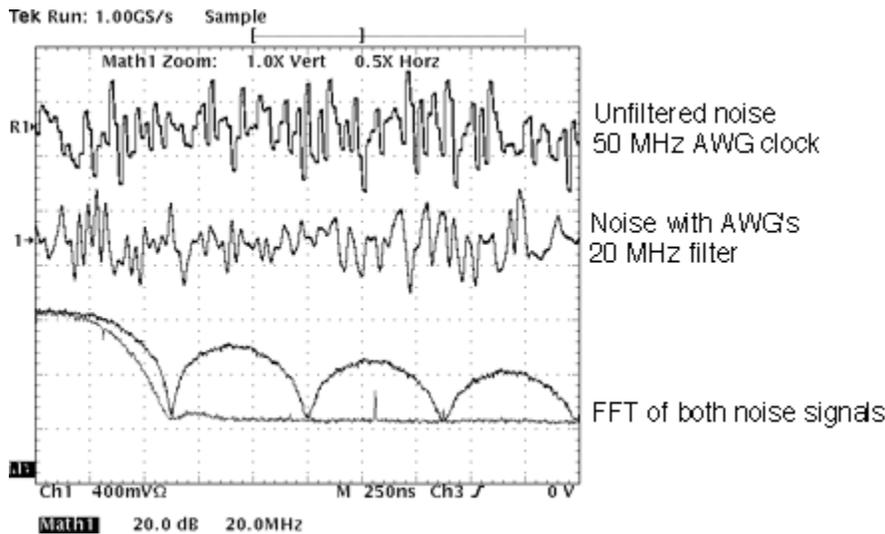


Figure 21. Upper waveform is a pseudo-random sequence with a 50 MHz clock. The second waveform is the same signal filtered by the AWG's 20 MHz low-pass filter. The lower two traces are the calculated FFT spectra. Note that the horizontal scale of the spectra is 20 MHz/div.

* In this paper, AWG always refers to Arbitrary Waveform Generator, as opposed to Additive White Gaussian, an unfortunate coincidence in this context.

Verifying Read Channel Frequency Response

A popular application of a noise source is to quickly verify the frequency response through various points in the read channel chain. The top trace in Figure 22 shows a 20 MHz bandpass filter's output response to an AWG noise waveform. The FFT of the filter output is shown in the second trace. If the AWG noise waveform had a constant noise density, then we could simply use the FFT of the output signal as our indicator of the channel's response. But the AWG noise waveform has its characteristic downward slope starting at DC (third trace).

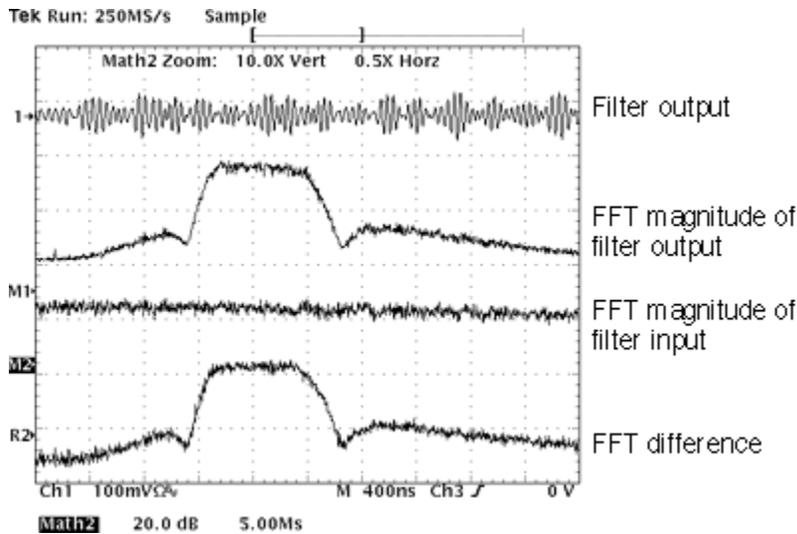


Figure 22. The top trace is a bandpass filter's response to an AWG noise waveform. The AWG noise signal was clocked at 1 GS/s and low-pass filtered at 100 MHz. The second trace is the FFT of the output signal. The third trace shows the downward sloping FFT magnitude of the clocked noise waveform. The TDS calculates the actual filter response by subtracting the FFT of the filter output from the FFT of the noise stimulus.

Thus, to correct for the 4 dB drop in noise density across the DC-to-50-MHz frequency span of interest, we must use the TDS' math function to subtract the filter's output spectrum from the noise input spectrum (bottom trace).

Figure 23 compares the calculated filter response with the response as measured by a conventional swept network analyzer. While the swept analyzer offers superior resolution and higher dynamic range, it is a costly, specialized tool that is not always available. The AWG/TDS combination is an effective alternative for quickly verifying channel filter performance.

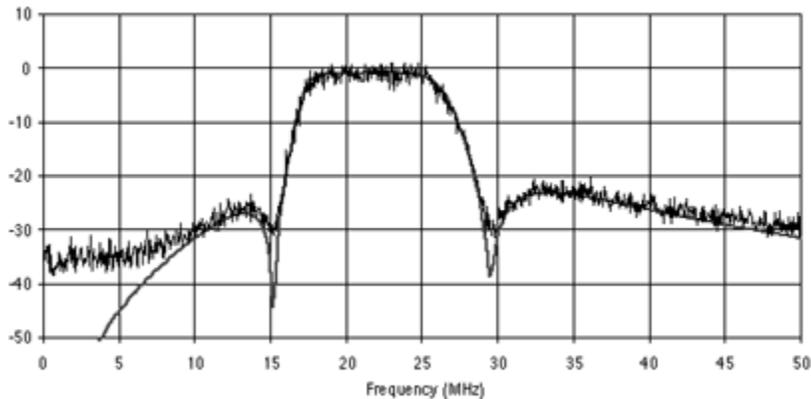


Figure 23. A comparison of TDS calculated filter response (from Figure 22) using the AWG's noise function against the response using a conventional swept network analyzer (smooth line).

Simulating Signal-to-Noise Ratios

The AWG's noise waveform can be added to our read channel waveforms to simulate different signal-to-noise ratios. The first step is to create a waveform record using the AWG's noise function. In Figure 24, a 960-point noise record is generated, which we will add to the 960-point 1F pattern shown in Figure 15.

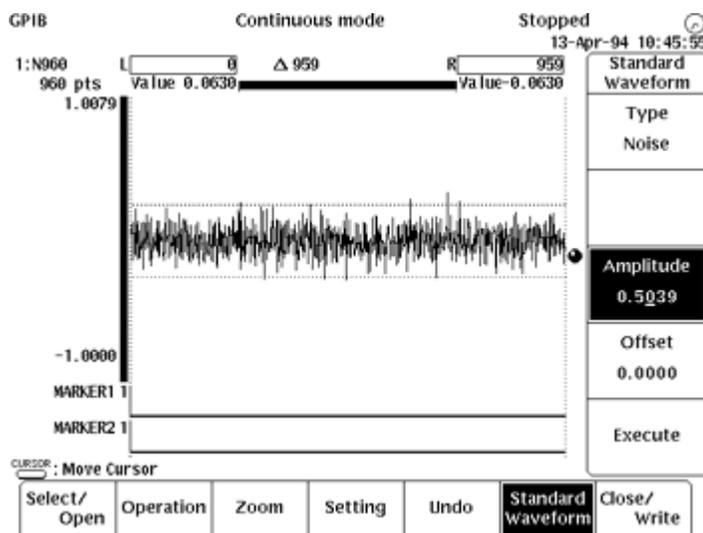


Figure 24. The noise waveform is specified in terms of its peak amplitude.

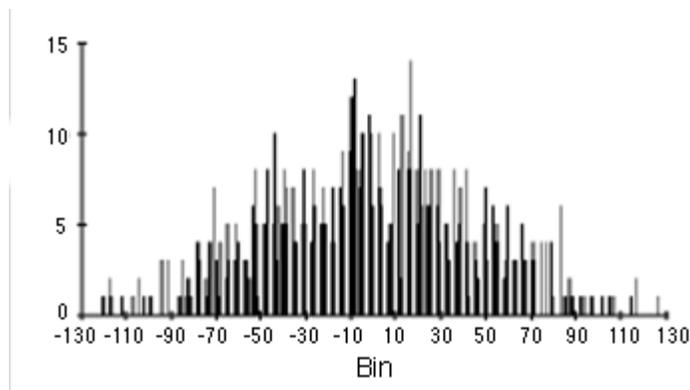


Figure 24a. The histogram of the 960-point noise waveform sorted into the 256 bins. Although a random number generator produces uniformly distributed values, the AWG processes the output of its random number generator so that its characteristics more closely match a Gaussian noise source.

As mentioned previously, the AWG simulates a noise waveform by calculating a sequence of random numbers. Digital random number generators typically produce uniformly distributed values, but circuit noise is better modeled with a Gaussian distribution. In practice, the AWG actually calculates a noise value by averaging 12 consecutive random numbers. Thus, by the central-limit theorem, the noise values will more closely approximate a Gaussian distribution than the underlying uniform distribution. Figure 24a shows a histogram of the 960-point noise waveform. There are 256 bins ranging from 127 to +128, each corresponding to a level of the AWG's 8-bit D/A converter. The histogram increasingly approaches a Gaussian distribution as the record length increases. Unlike a true normal distribution, note that the tails of the AWG noise distribution abruptly cut off at the specified peak value. This peak value is entered as the amplitude of the noise waveform. The crest factor, or peak to RMS ratio, is approximately 3 to 1.

Figure 25 shows the resulting waveform as displayed by the AWG's waveform editor.

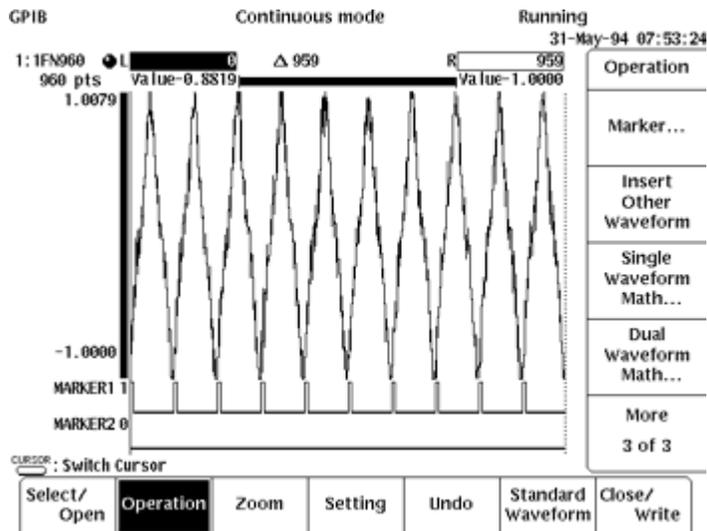


Figure 25. The 960-point noise waveform is added to the 960-point 1F waveform using the AWG's dual waveform math function. The result is a 960-point waveform. Marker pulses were added to provide a convenient trigger pulse once per cycle.

Making Coherent Noise More "Random"

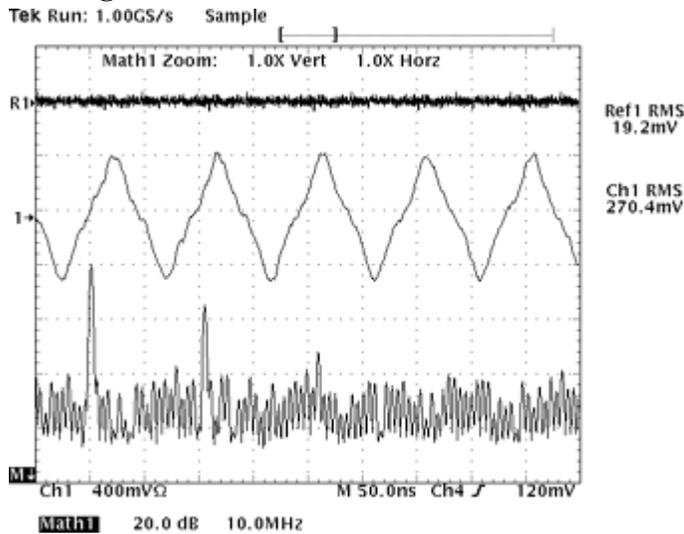


Figure 26. The top trace is the 960-point noise waveform (19.2 mV rms). The middle trace is the 1F waveform with noise (270 mV rms). The bottom trace is the FFT of the waveform.

Figure 26 shows the 960-point noise waveform and the 1F waveform plus noise as captured by the TDS. The wideband signal-to-noise ratio is $20 \cdot \log(270/19)$ or 23 dB. The bottom trace is the FFT of the waveform which gives us more revealing information. While the ~ 10 MHz 1F fundamental and its odd harmonics are the dominant components, there are clearly discrete spectral components rather than the desired flat noise floor. The problem lies in the selection of the record length for the noise waveform. As discussed in the beginning of this section, a 960-point noise waveform will repeat every 960 points. In this case, the AWG clock rate was 1 GS/s, making the noise pattern repeat every 960 ns or at about 1 MHz. Furthermore, the noise pattern is pre-calculated with the 1F pattern such that the two waveforms are forever coherent. This is an unrealistic test signal to simulate a noisy channel.

With a pre-calculated signal plus noise waveform, we cannot completely escape the fact that the two components are implicitly correlated. We do not want to change the fundamental 1F frequency of ~ 10 MHz since this is the signal component. We can, however, reduce the effect of the correlation between the signal and noise components by decreasing the repetition rate of the noise waveform. For example, we can simply multiply the record lengths by 250 so we have 240,000-point waveforms instead of 960-point waveforms. The new 1F record is the same fundamental signal repeated over a longer interval. But the 240,000-point noise waveform repeats 250 times less often than the 960-point noise waveform clocked at the same rate.

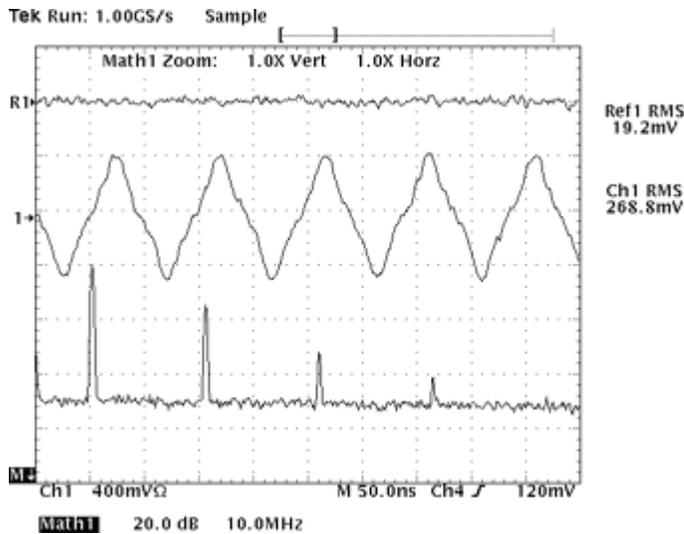


Figure 27. When the waveform record length is increased to 240,000 points, the noise density better approximates a level noise floor. The signal-to-noise ratio is actually the same (23 dB), but the longer record length provides a more realistic test signal.

Figure 27 shows that the discrete spectral noise components have been transformed into a more recognizable noise floor, and we now have a more suitable test signal. The longer noise record is still a repetitive waveform, but its noise density characteristics are better distributed in the frequency domain. The TDS' FFT measurement in Figure 26 and Figure 27 were derived from a time domain record sampled at 1 GS/s and a record length of 5000 points. Thus, each FFT calculation is based on a 5 μ s snapshot of the test signal, but the noise pattern repeats only every 250 μ s. This means that we need to average the FFT calculations over time to accumulate its characteristics. The bin size, or spacing between FFT frequency domain samples, is 200 kHz, or the ratio of the sample rate to the record length. The equivalent noise bandwidth of the FFT filter was about 2 bins. Thus, when averaged over time, the noise density at this frequency resolution is more evenly distributed with the longer record.

8. Inserting Timing Impairments

In addition to amplitude and noise impairments, timing impairments represent a third class of variations that affect the basic channel waveform. Timing variations inserted into the channel waveform simulate the jitter or peak shifting that results from a variety of mechanical and electrical sources within a drive. For example, instantaneous spindle speed variations during writing or reading produce jitter, while poor channel equalization circuitry causes data dependent peak shifts. The AWG provides a variety of techniques to control the precise amount and location of an impairment in order to stress or characterize specific channel functions, such as clock recovery performance.

Simulating Peak Shift (Jitter)

The most commonly used timing impairment is peak shift. The objective is to move the location of one or more peaks forward or backward relative to other peaks. At first glance, this may seem to be a trivial exercise solved by a simple cut-and-paste operation. That is, why not just move a selected segment of a waveform forward or backward in the record? Figure 28 illustrates this approach. An approximately 20 ns segment of a modulated data waveform is simply advanced in time by 1 ns, or one record point for a 1 GS/s clock. The peak shifts 1 ns, but this creates two abrupt transitions at the splice points. In this example, the splices are placed where the signal slopes are large to illustrate that simple shifting can introduce discontinuities resulting in an unrealistic test signal. You can attempt to hide the splice by selecting splice points at peak locations or where the slope of the signal is zero. But this simply masks the fact that the altered signal is not a realistic implementation of peak shift. Remember that our model of a disk waveform is a sum of transition responses where the effect of a single write transition extends beyond its local peak. Thus, in a realistic test signal you cannot alter one peak location without accounting for the effect on adjacent peaks.

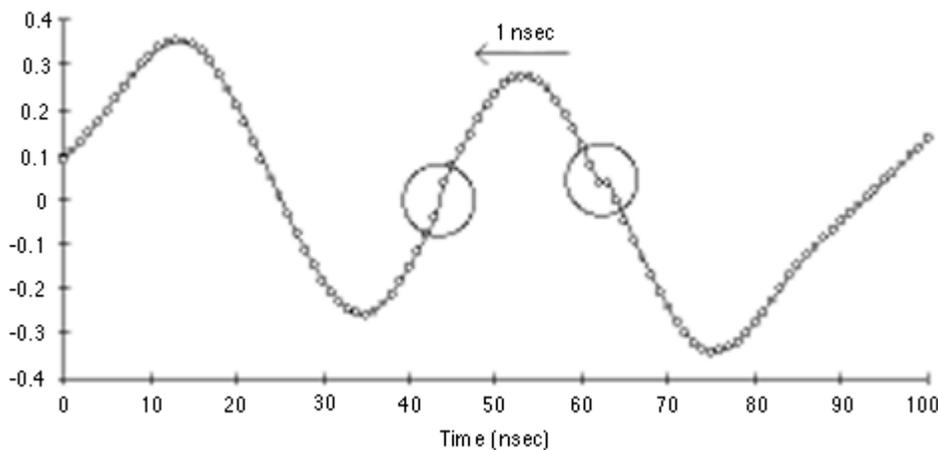


Figure 28. The second positive peak is advanced by simply cutting a 20 ns slice and pasting it 1 ns earlier in time. This introduces discontinuities at the splice points and skirts the realities of the read-write process.

The solution is to simply re-convolve the transition response waveform with an altered data pattern. In particular, the data transition corresponding to the desired peak is simply advanced by one location to simulate a 1 ns shift in the location of the write-current transition. Recall that for our data pattern example, each bit interval is 20 ns long, consisting of 20 repeated 1's or 0's for a 1 ns clock period. An extra "1" replaces the last "0" from the preceding bit interval thus advancing the location of the positive-going transition by 1 ns. Figure 29 shows the original data waveform and two successive re-

convolved data patterns where the positive-going transition was advanced and delayed by 1 ns. The peak shift is smoothly integrated into the waveform. Since our model of the transition pulse response was 256 ns long, we should expect that any transition should affect an interval tens of nano-seconds before and after the actual peak location. Note that this technique is limited to shifts that are integer multiples of the clock period. We will examine sub-period shifts later in this section.

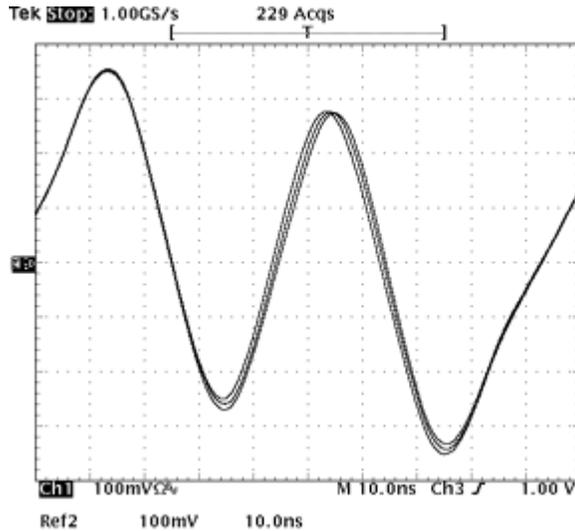


Figure 29. A single transition in the data pattern was advanced and delayed by 1 ns. The two new data patterns were convolved with the transition response waveform. The two new waveforms are compared against the original waveform. Although the location of a single transition was altered, the two adjacent peaks are also affected. The scope was triggered by an AWG marker output located at the same point in each waveform.

Simulating Frequency Shift to Verify Clock Recovery Loops

A second common timing impairment is frequency shift. Frequency shifts are especially useful for verifying acquisition and tracking of clock recovery loops. The most direct method of modulating the clock rate of the AWG is via its external clock input, receiving modulation from an external signal generator. That is, arbitrarily shifting the frequency of a complex waveform that is already modulated by the data pattern is best performed by an external clock generator. However, you can directly implement simple shifts, such as frequency modulation of a sinusoid, using only the AWG. Figure 30 illustrates another simple case of frequency shift modulation of a data pattern using just the AWG.

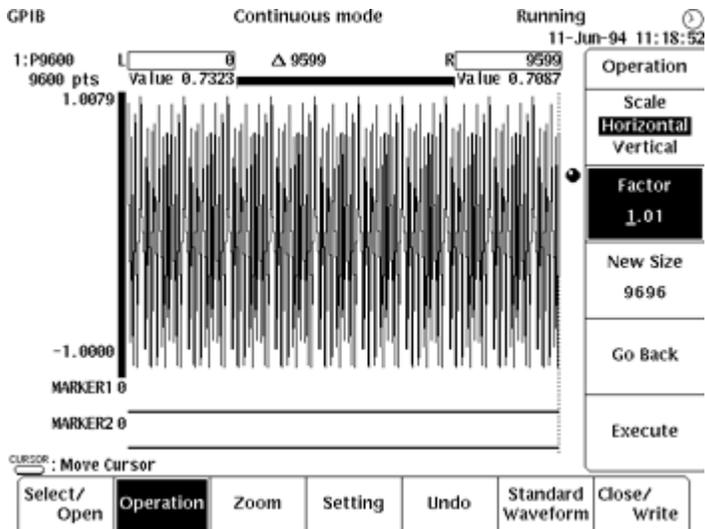


Figure 30. The AWG's horizontal scaling function is used to expand the 9600-point waveform by 1%, making the new waveform 9696 points. If the same 1 GS/s clock is used, the new waveform is 1% slower than the original.

The objective is to use our basic data waveform but to alternate its rate by $\pm 1\%$. We will use the AWG's horizontal scaling function to create two new waveforms--one that is 1% longer and one that is 1% shorter than the original. The original record length is 960 points, so we need to add and subtract 9.6 clock periods to the record length. Since fractional periods are not allowed, the simplest approach is to use the sequence editor to concatenate 10 copies of the original waveform and create a 9600-point waveform. Then we scale the new waveform to fit 9696 points and 9504 points to implement the frequency shift. Figure 30 shows the horizontal scaling function with a factor of 1.01 to expand the 9600-point waveform to 9696 points.

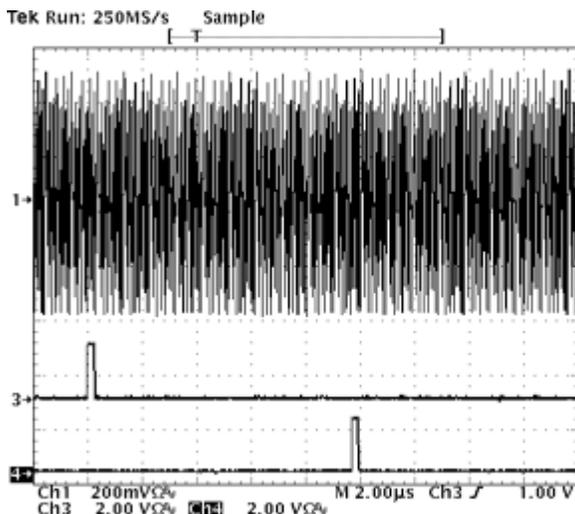


Figure 31. The 1% fast and 1% slow waveforms are concatenated to create a data pattern (Ch1) that toggles between two data rates.

Figure 31 shows the result as viewed on the TDS. We took advantage of the two independent marker outputs of the AWG to generate separate trigger pulses to signify the start of each direction of frequency shift. Use the marker triggers to synchronize the scope when examining the error signals in the clock recovery circuit. You can take advantage of the logic triggering capability in the TDS to overlay and examine the alternating patterns. In Figure 32, the TDS is set to trigger when either marker input goes high, so the scope captures the starting point of both patterns. The peaks diverge from the splice point since a constant frequency difference integrated over time causes a linearly increasing timing shift.

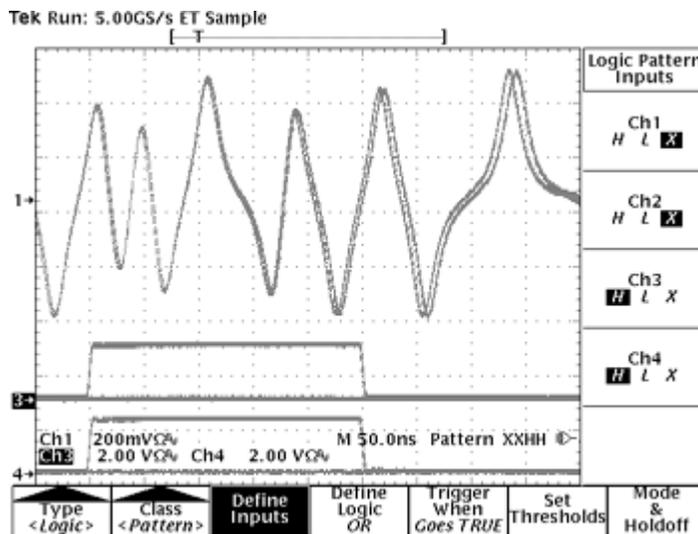


Figure 32. The TDS' logic trigger captures and overlays the slow and fast waveforms. The logical "OR" function is enabled so that the TDS will be triggered on the rising transition of either Ch3 or Ch4, which are connected to the AWG's Marker 1 and 2 outputs.

The AWG's two independent marker output signals, the lower two traces, were programmed to indicate the alternating rates; they signify each change of direction in frequency. Ch3 marks frequency decreasing by 1%; Ch4 marks frequency increasing by 1%.

"Beating the Spec" to Achieve Higher Resolution

The previous examples altered timing by changing the AWG's record in increments of the clock period. This limits the timing resolution to 1 ns for a 1 GS/s AWG clock. The next

two examples illustrate techniques to achieve timing shifts with resolutions better than 1 ns, surpassing the specified timing resolution of the AWG. The first example demonstrates how to alter the transition pulse response equation to shift a waveform. The second example shows how to shift a selected segment of an existing waveform. Recall that the nominal location of the transition pulse peak was defined by the 128 ns offset in the pulse equation. Figure 33 shows the AWG's equation editor with the timing offset changed to 127.8 ns. This means that when our data pattern is convolved with this pulse response, each peak will arrive 200 ps earlier. If the offset is changed to 128.1 ns, then each peak in the convolved waveform will arrive 100 ps later than in the original waveform.

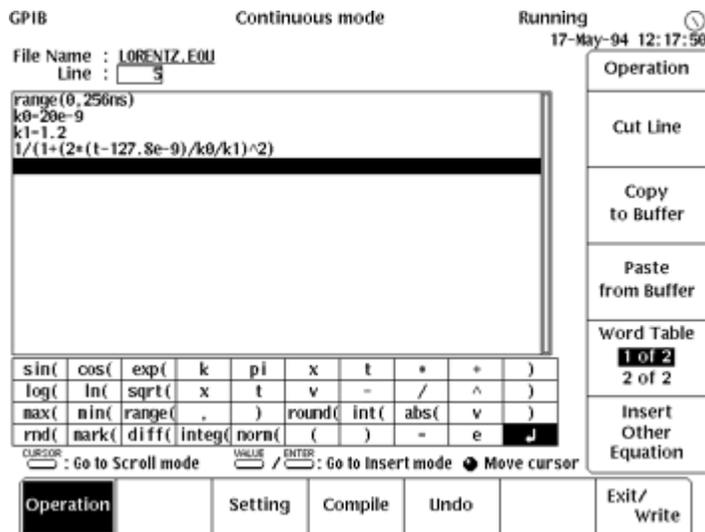


Figure 33. The timing offset in the transition response equation is changed from 128 ns to 127.8 ns. When convolved with the original data pattern, the resulting waveform is mathematically advanced by 200 ps.

Figure 34 demonstrates the result. The original 48-bit data pattern was re-convolved with four additional transition responses, having offsets of 200, 100, +100, and +200 ps relative to the original offset of 128.0 ns. The TDS displays the five resulting waveforms. The scope was triggered by an AWG marker output pulse stored at the same location in each waveform.

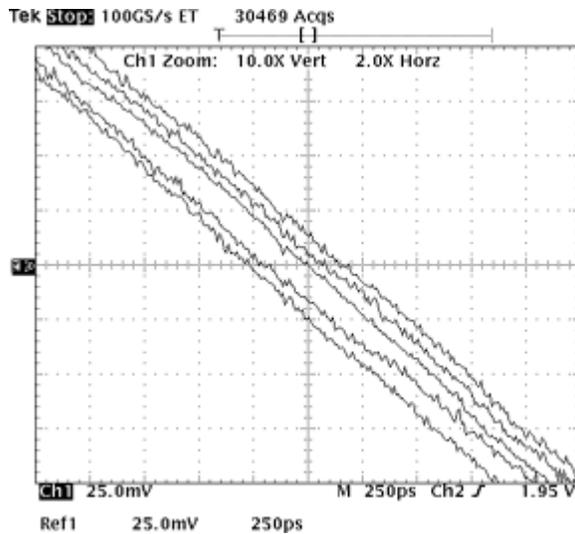


Figure 34. The TDS displays a single edge of the original data waveform and four other waveforms with 100 ps increments of timing offset in the transition response equation.

The waveforms are vertically expanded to isolate the same negative-going zero-crossing. Note that the horizontal scale is 250 ps per division. Ideally, the waveforms would be equally spaced in 100 ps horizontal increments, spanning a total of 400 ps. Instead, the increments are not uniform and the total span is about 450 ps. Why? The reason is that while the mathematical calculation of the convolved waveforms is precise, we are limited by the resolution and linearity of the AWG's 8-bit D/A converter. Thus, when using this technique, you need to examine the results, and you may need to adjust the timing offset empirically to achieve a specific output shift.

The final example addresses the task of a sub-nanosecond shift of a segment of an existing waveform. In other words, you may have captured an actual drive signal with the TDS and transferred it to the AWG. In this case, you do not have the option of applying various transition pulse response functions with the convolution function. We will use the technique introduced in the first example of this section. That is, we will select a segment of the waveform and shift it in time. Because we are shifting the segment by only a fraction of a clock period, we can ignore the discontinuities introduced at the splice points.

In summary, we will select a segment and scale it to ten times its original length. The expanded waveform is shifted by two points and then collapsed to its original length. The net effect is a shift of only 0.2 points, or 200 ps for a 1 ns clock period. In Figure 35, the AWG's vertical cursors isolate a forty point segment (points 503 to 542) of our 960-point basic data waveform. The AWG's horizontal scaling function is used to expand this segment by a factor of 10 to 400 points.

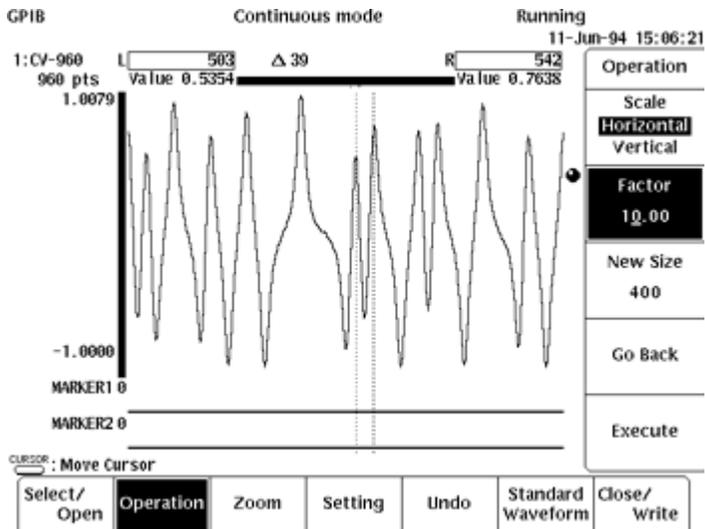


Figure 35. The AWG's horizontal scaling function will expand a 40-point segment between two positive peaks by a factor of 10 to 400 points.

After scaling, the AWG's shift function moves the expanded segment forward in time by 2 points (Figure 36). The two original leftmost points in the segment are lost, and we defined the two new rightmost point values to be copies from the original rightmost point value. The AWG's horizontal scaling function is then re-applied with a factor of 0.1, so that the expanded segment is reduced to its original size of 40 points.

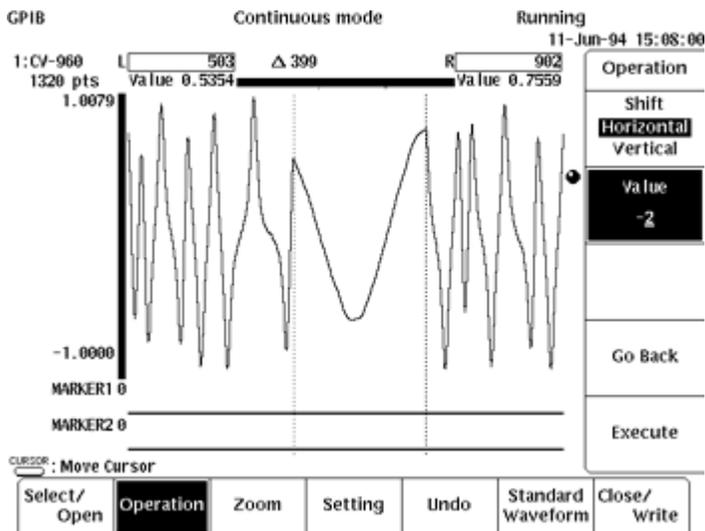


Figure 36. The 40-point segment has been expanded to a 400-point segment. Then the AWG's horizontal shift function moves each data point forward in time (leftward) by 2.

Figure 37 illustrates this procedure. The filled points in the lower waveform represent the original waveform. The unfilled points represent the expanded version with ten times the points of the original. The expanded signal is shifted left by two points at the expanded scale, and the shifted waveform is scaled back to its original size.

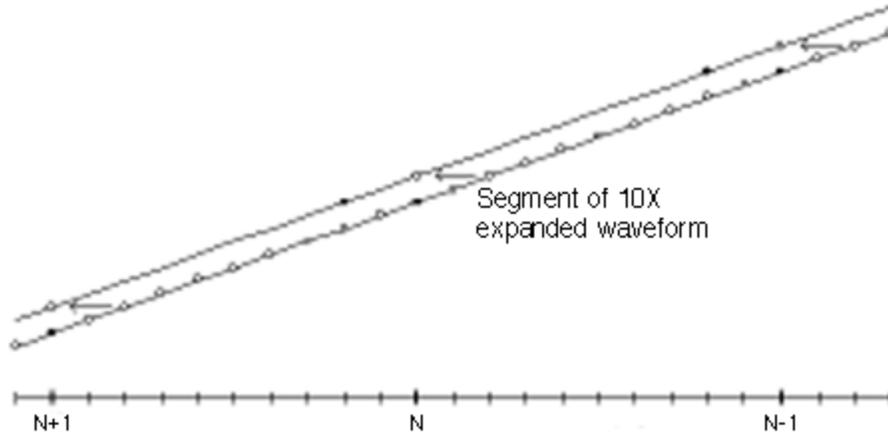


Figure 37. The filled points on the lower waveform are interpolated to simulate a tenfold increase in the timing resolution. The expanded waveform is shifted left by two points at the expanded scale. When the expanded waveform is scaled back (decimated) to its original size, the waveform has been shifted left by two tenths of a point.

Figure 38 shows the results as captured by the TDS. The expand-shift-contract process was done twice--once to advance the segment by 200 ps and once to delay the segment by 200 ps. The TDS captures the two new waveforms as well as the original waveform. Note that the selected segment spanned two consecutive positive peaks. The TDS zoom function was used to expand the three waveforms around the beginning of the affected segment. The three waveforms overlay nicely before the affected area, but they diverge in time as they enter the 40-point (40 ns) segment. Although the ideal result of three equally spaced waveforms, 200 ps apart, is not achieved, we have transcended the AWG's resolution specification. The outcome is repeatable and meets the objective of improving on the apparent "limitation" of a 1 ns timing resolution for a 1 GS/s AWG clock.

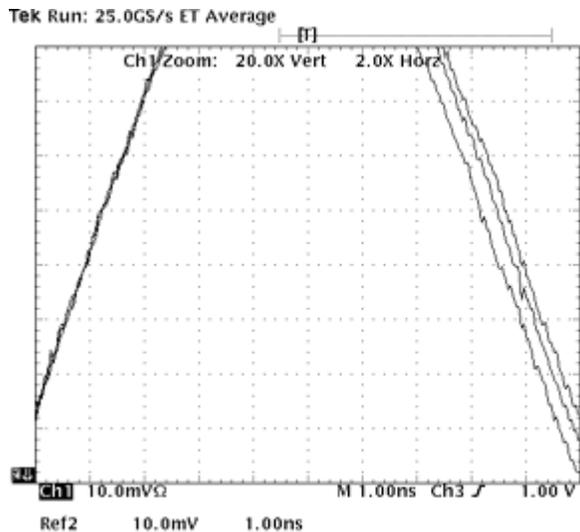


Figure 38. The segment to the right of the positive peak (above the top of the screen) was advanced and delayed by 200 ps relative to a fixed marker output sync pulse.

9. Appendices

Related Tektronix Publications

- [TDS 744A Digitizing Oscilloscope Data Sheet](#)
- [TDS FFT Application Note](#)
- [AWG 2041 Data Sheet](#)
- [AWG 2021 Data Sheet](#)

Addendum -- Region Shift

Section 8 of this document describes a method for simulating jitter by convolving the data pattern with a second, slightly modified version of itself. This technique applies to the existing AWG 2000 Series products: AWG 2005, AWG 2021, and AWG 2041.

Just before press time, Tektronix announced a new feature, known as "Region Shift," for AWG 2000 Series instruments. The first product to incorporate the feature is the AWG 2021.

Region Shift allows the user to automatically modify a waveform's placement in time. The technique applies a minute amplitude shift to each point in a selected waveform region, with the net effect of making each point appear earlier or later in time relative to the unmodified pulses surrounding it. Using Region Shift, the AWG 2021 can produce jitter excursions as small as 4 ps.

AWG 2000 Series Arbitrary Waveform Generators

Tektronix AWG Arbitrary Waveform Generators give you the most extensive capabilities for editing waveforms, with 8 or 12 bits of vertical resolution and waveform frequencies to 500 MHz. AWGs contain a high speed, high resolution DAC with sophisticated triggering and mode settings, plus up to 4 megabytes of internal waveform memory in which to create and edit waveforms.

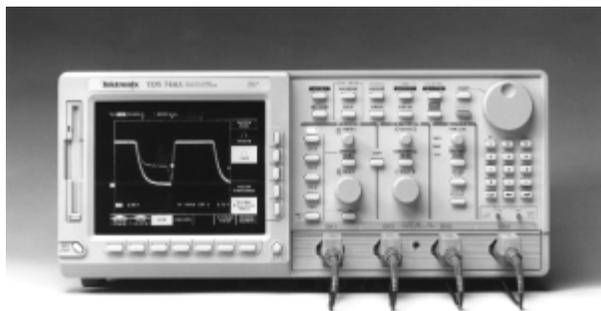


You can import a waveform from a scope or disk, select one of the AWG's standard waveforms, or define your own from scratch. Using powerful built-in editing tools, you can edit the waveform right on the screen, then clock it out at rates up to 1 GS/s.

	AWG 2041	AWG 2021
Clock Rate	1 GS/s	250 MS/s
Max Output Waveform Frequency	500 MHz	125 MHz
Region Shift	No	Yes
Direct Waveform Transfer from Digital Scopes/Digitizers	TDS Series, 2000 Series, 11000 Series, DSA, RTD 700 Series, and others	Same as AWG 2041
External Clock	Clock In & Out	Clock In & Out
Graphical Waveform Editing	Draw, Timing, Table, Equation, FFT (Opt. 09), Digital Word/Pattern Generator (Opt. 03)	Same as AWG 2041

Max Peak-to-Peak Amplitude into 50 ohms	2 V	5 V
Memory: Execution per Channel	1 MB; expandable to 4 MB (Opt. 01)	256 kB per channel
Memory: Non-volatile	512 kB	512 kB
Modulation	AM	AM
Output Channels	1 Analog 8 ECL Digital (Opt. 03)	1 Analog 2 Analog (Opt. 02); 12 ECL Digital (Opt. 03); 24 TTL Digital (both Opt. 02 and Opt. 04)
Sine, Square, Triangle, Ramp, DC, Pulse	Synthesized only; 10 MHz	Synthesized only; 2.5 MHz
Sweep	Sequencer and Equation Editor can be used to create sweep	Same as AWG 2041
Time Base Accuracy	1 ppm	50 ppm
Vertical Resolution	8 bits	12 bits
Built-In Floppy Drive (3½" HD PC-Compatible)	Yes	Yes

TDS 744A Digitizing Oscilloscope



The TDS 744A represents the next generation of digitizing scope performance. This versatile general-purpose instrument introduces Tek's new Instavu™ acquisition feature and sets a benchmark in waveform capture rate for DSOs. The TDS 744A can display more than 400,000 acquisitions per second--a rate 2,500 times faster than the most advanced DSOs available.

Other TDS 744A features include advanced triggering, graphical user interface, high-speed signal processing, complementary probing, and sophisticated documentation capability.

With its long record length and high bandwidth, the TDS 744A is an ideal complement to the 2000 Series AWGs for disk drive testing. Its feature set is also suited to applications ranging from digital system development to manufacturing test.

	TDS 744A
Bandwidth	500 MHz
Input Channels	4
Sample Rate per Channel	
1 channel	2 GS/s
2 channels	1 GS/s
3 or 4 channels	500 MS/s
Vertical Resolution	8 bits; > 12 bits with Hi-Res; 11 bits with Averaging
Record Length	500 to 50,000 points per channel Max. 500,000 points (optional)
DC Gain Sensitivity	± 1.0%
Vertical Sensitivity	1 mV to 10 V/div; 50 ohm: 1 mV to 1 V/div
Automatic Measurements	25
Triggering System	Edge, Pulse (Width, 1 ns Glitch, Runt, and Slew Rate), Logic (Pattern, State, and Setup and Hold Time Violation), HDTV Video (optional)
Special Features	InstaVu, Dual Window Zoom, FFT, Differentiation, Integration,

Color Monitor, 3½" Floppy Drive