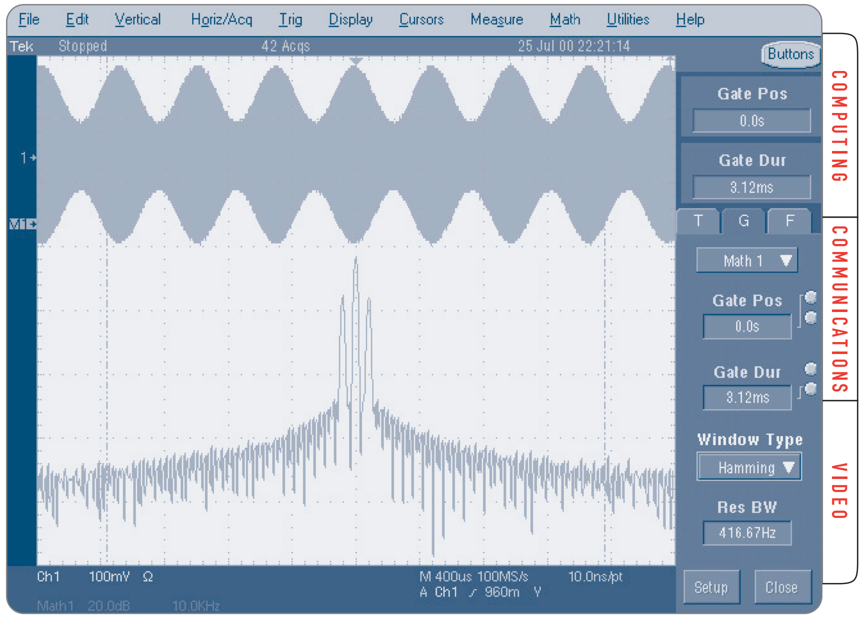


Spectral Analysis: Analyzing a Signal Spectrum

Introduction

The TDS7000 Series DPO (digital phosphor oscilloscope) provides many powerful capabilities for analyzing the spectral properties of a signal over an extremely broad range of frequencies.

In fact, the oscilloscope's capabilities equal, and in many cases exceed those of specialized spectrum analyzers. Built-in capabilities include amplitude and phase spectral displays, spectral windowing, and measurement of numerous practical waveform characteristics. For more in-depth analysis, sampled data can be further processed using on-board analysis tools such as Excel and MATLAB. This capability significantly expands and extends the functionality of the oscilloscope.

This application note begins with a brief tutorial covering basic spectral analysis concepts. Next, it describes how to use the TDS7000 Series oscilloscope to analyze a signal spectrum, and uses examples to describe how the screen displays relate to fundamental spectrum concepts. Finally, examples are given that illustrate the use of some of MATLAB's powerful analysis tools to process captured signals.

Tektronix
Enabling Innovation

Fundamentals of Spectrum Measurement

Power spectrum estimates describe how signal power is distributed across frequency given a finite record of the input signal. It is useful to consider deterministic and random signals as two separate cases. The respective techniques are similar but do have distinct differences. Both types of signals are commonly encountered in many engineering problems. This section discusses the fundamentals of spectral estimation and describes some of the commonly used methods. The oscilloscope of the discussion is restricted to real signals (i.e. not complex) and to estimating the power spectrum of continuous-time signals from sampled data.

First, define the oscilloscope's sampling rate and sampling period as f_s and T_s, respectively, where $T_s = 1/f_s$. The continuous-time input signal, for which the spectrum is to be computed, is denoted as $x(t)$. The sampled signal is $x[n]=x(nT_s)$ for $n = 0,1,\ldots,(N-1)$, where N is the record length. The time variable is t with units of seconds, the discrete time variable is n with units of samples, and the frequency variable is f with units of Hz.

The Discrete Fourier Transform (DFT), which is implemented using a Fast Fourier Transform (FFT) algorithm, is a widely used tool in spectral measurement. The DFT is defined as

$$X(f_k) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/K} \qquad k = 0,1,\ldots,(K-1)$$

where

$$f_k = {kf_s}/{K},$$

and K is the number of points in the frequency domain. The DFT is often defined with K = N. However, using K>N is more general and can be implemented by appending zeros to the end of the data record (called "zero padding"). This case will be assumed here since the TDS7000 Series uses zero padding to provide more spectral display points.

Power spectrum measurements of random signals such as noise or interference are discussed first. Assume that the essential characteristics (such as mean, variance, etc.) are constant over the finite time interval. The mathematical definition of the power spectrum for a continuous-time random signal is:

$$P(f) = \lim_{T\to\infty} E\left\{ \frac{2}{T} \left| \int_{-T/2}^{T/2} x(t) e^{-j2\pi ft} dt \right|^2 \right\}, \text{ Watts/Hz}$$

where $E\{*\}$ is the ensemble average of all possible realizations of the random signal. Note, the units are Volts^2/Hz or, equivalently, Watts/Hz assuming $x(t)$ is the voltage across a one Ohm resistor. In other words, the estimate is scaled such that the signal power contained within a given frequency band is the area under $P(f)$ over the frequency range of interest. If the signal is sampled at a rate greater than two times the maximum frequency of the signal (i.e., the Nyquist rate), then the power spectrum can be approximated using the DFT as:

$$P(f_k) = \lim_{N\to\infty} E\left\{ \frac{2T_s}{N} |X(f_k)|^2 \right\} \qquad 0 \le f_k \le {f_s}/{2}.$$

This function is difficult to evaluate in practice, and usually must be estimated from a finite-length data record. Looking at the above equation, the simplest approximation of the power spectrum from a record of length N is:

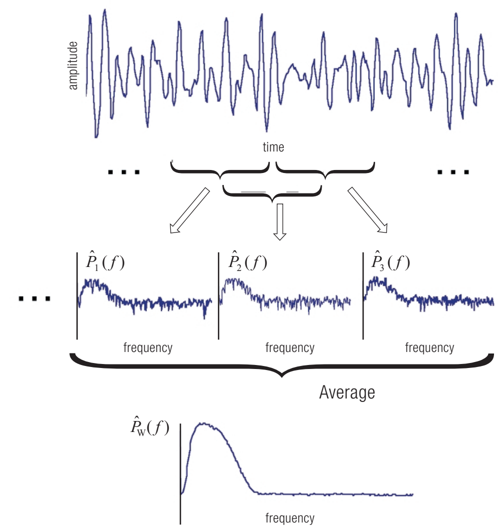
$$\hat{P}(f_k) = \frac{2T_s}{N} |X(f_k)|^2 \qquad 0 \le f_k \le f_s/2 .$$

This simple estimate of the spectrum is frequently improved by pre-multiplying the data by a window function: $\tilde{x}[n] = w[n]x[n]$. An excellent discussion of window functions and their characteristics is given in the TDS7000 Series manual. The effects of the window are included in the estimate as follows:

$$\hat{P}(f_k) = \frac{2T_s}{NU} |\tilde{X}(f_k)|^2 \qquad 0 \le f_k \le f_s/2$$

where $U = \frac{1}{N}\sum_{n=0}^{N-1} w^2[n]$ is the coherent gain correction for a given window and $\tilde{X}(f_k)$ is the DFT of $\tilde{x}[n]$.

This estimate of the power spectrum is called a simple periodogram. Because it does not include any averaging, the variability in the estimate is relatively high. The variance of the estimate has been shown to be approximately on the order of the square of the spectrum itself. Furthermore, the frequency resolution is inversely proportional to the record length and also depends on the type of window function.



▶ ***Figure 1.** Steps for computing averaged spectrum.*

If the underlying characteristics of the signal are not changing with time, a direct approach to reduce the variability of the estimate is to do some averaging. This can be accomplished recording M consecutive data segments, computing a simple periodogram of each segment, and averaging the simple periodograms. The segments can be overlapping for additional smoothing. Figure 1 graphically shows the steps in computing this averaged periodogram, which is known as the Welch averaged periodogram spectrum estimate. The equation for computing the averaged periodogram is:

$$\hat{P}_W(f_k) = \frac{1}{M}\sum_{m=1}^{M} \hat{P}_m(f_k) \qquad 0 \le f_k \le f_s/2$$

where $\hat{P}_m(f_k)$ is the simple periodogram of the m^{th} segment of data,

$$\hat{P}_m(f_k) = \frac{2T_s}{LU} |\tilde{X}_m(f_k)|^2 \qquad 0 \le f_k \le f_s/2$$

and L is the record length of each individual segment.

The Welch periodogram is reasonably computationally efficient due to the use of FFT algorithms, and provides good performance for many types of signals. Consequently, it is widely used and is often incorporated into analysis tools such as MATLAB.

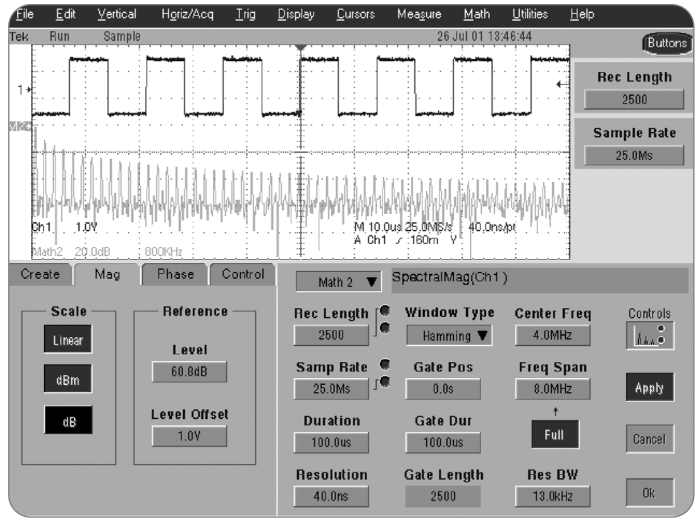
To use the Welch periodogram, a number of parameters must first be defined. These include the number of segments (M), the length of each segment (L), the amount of overlap between segments, and what window function to use. In making these decisions you face a tradeoff between frequency resolution and the variability in the spectral estimate. When longer segments are used, frequency resolution is better but fewer segments are available for averaging and smoothing. On the other hand, shorter segments yield lower resolution but result in much smoother estimates.

Using overlapping segments increases the number of segments to be averaged, but the overlapping segments contain the same information so the reduction in the variability of the estimate is limited. A typical amount of overlap is 50%. The choice of the window also impacts the spectral estimate giving more control over the resolution characteristics of the spectral estimate.

As already mentioned, analysis of periodic deterministic signals differs somewhat from that of random signals. As with random signals, analysis is based on the DFT and FFT, and the power spectrum estimate in units of Watts (assuming a one ohm load) is given as:

$$\hat{P}(f_k) = \begin{cases} \dfrac{2}{N^2 U}\left|\widetilde{X}(f_k)\right|^2 & 0 < f_k \le f_s/2 \\ \dfrac{1}{N^2 U}\left|\widetilde{X}(f_k)\right|^2 & f_k = 0 \end{cases}$$

where $\widetilde{X}(f_k)$ is the DFT of the windowed signal $\widetilde{x}[n]$. Here, the amplitude of the harmonic peaks in the spectrum corresponds to the mean square value of each harmonic component. Note the similarity between this deterministic case and the simple periodogram for the random case, the only difference being the scale factor. Finally, it is important for the time gate to record as many cycles of the periodic signal as possible in order to closely approximate the harmonic spectrum.



▶ ***Figure 2.*** *Oscilloscope screen showing a simple square wave and its corresponding spectrum magnitude.*

Spectrum Measurement Using the TDS7000 Series Oscilloscope

Now that the basics of spectral estimation have been covered, we next describe how the Tektronix TDS7000 Series oscilloscope can be used to investigate spectral properties of an input signal. This section gives several practical examples, and relates the results to the theoretical development in the previous section.

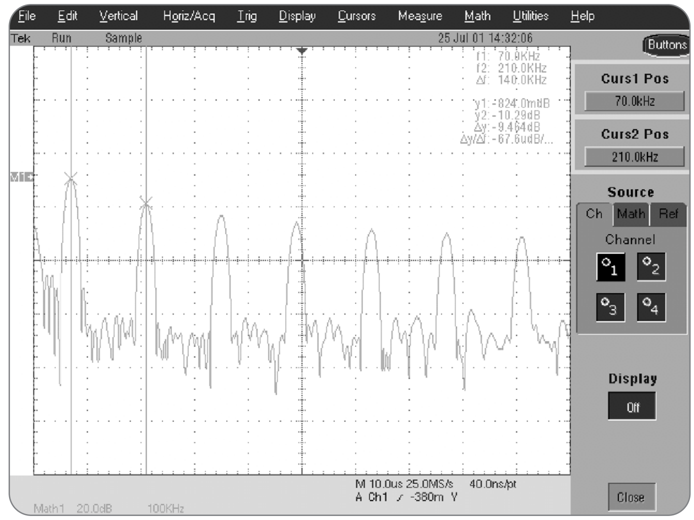
As a first example, we investigate the spectrum of a 70 kHz 2V (peak-peak) square wave with 50% duty cycle input on channel 1. A typical analysis of this periodic signal includes the following steps:

In the Math option, select Spectrum Setup to bring up the main menu (see Figure 2). Input a sampling rate in the “SAMP RATE” field, which determines the maximum possible analysis frequency, $f_{max}=f_s/2$. Use $f_s = 25$ Ms/s for this example, giving $f_{max} = 12.5$ MHz.

Next select record length using either time (e.g., seconds) or number of samples. Choose 2500 samples (100 µs). The oscilloscope will compute the spectrum using samples within a specified time gate. To use the entire record, set the gate position at 0 ms and the duration equal to the record length of 100 µs. The time gate can be adjusted to focus on particular aspects of the waveform, however spectral resolution decreases as the gate is shortened. Since multiple cycles of the square wave are included within the selected time gate, expect a line spectrum showing the harmonics of the periodic waveform.

Now select a window to reduce sidelobes introduced by the finite-length time gate. A "rectangular" window has the worst sidelobes and is seldom used. Use a Hamming window instead. (See Chapter 3, page 169 of the product manual for more information about windowing).

A center frequency of 4 MHz and a frequency span of 8 MHz are set as the initial values. The spectrum of Channel 1 is computed and displayed by selecting "Math 1", then clicking Magnitude under the Create tab, and then selecting Channel 1. A dB amplitude scale is the default. The square wave and the resulting spectrum are shown in Figure 2. More detailed analysis is possible by zooming in on a narrower frequency range. For example, setting the center frequency to 500 kHz and the span to 1 MHz gives the results shown in Figure 3. The data trace (channel 1) is turned off by clicking Math, Display On/Off, channel 1 under the Source Ch tab, and then clicking the Display button to select off.

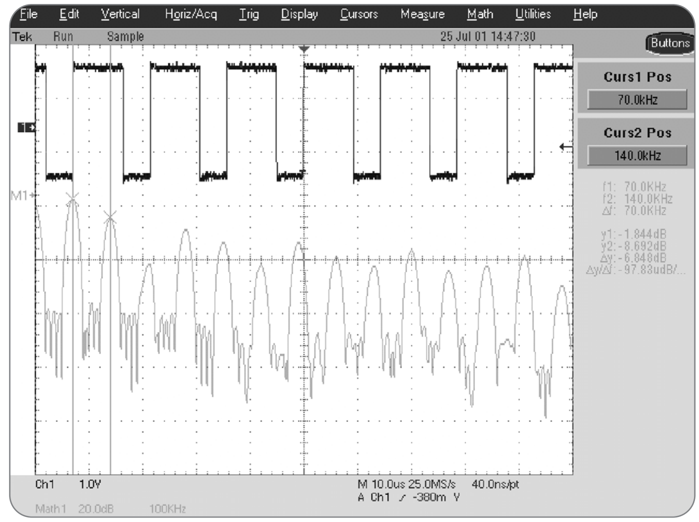


▶ ***Figure 3.*** *Oscilloscope window showing a zoomed-in view of the magnitude spectrum.*

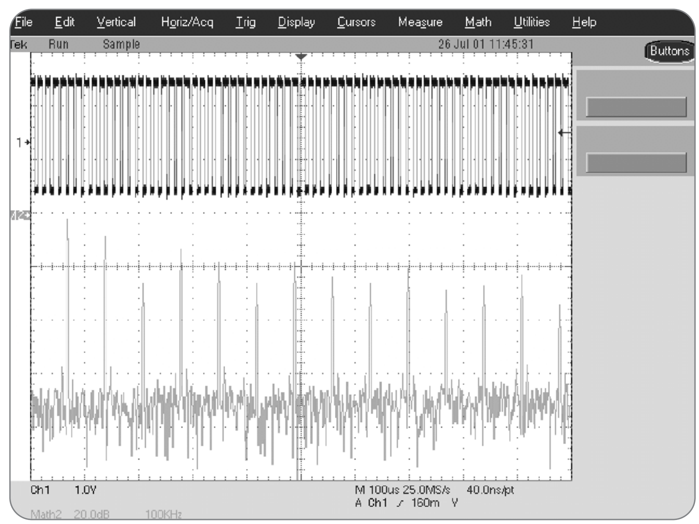
As expected for a 50% duty cycle square wave, spectral harmonics are seen at odd multiples of the 70 kHz fundamental frequency (70 kHz, 210 kHz, etc). Using the dB scale, the height of the k-th harmonic can be interpreted as the power of the sinusoid at that frequency relative to a reference value. The reference value is determined by the Level Offset parameter under the Magnitude tab. For example, if the Level Offset is 1V, then displayed power represents power in dB relative to 1 watt assuming a 1 ohm load. If the dBm scale is selected, the Level Offset value is automatically set to 223.6 mV. This value equals the rms voltage of a 1 mW signal into a 50 ohm load; therefore the display shows power above 1 mW assuming a 50 ohm load. Note that the scale difference between dBW with a 1 ohm load and dBm with a 50 ohm load is 10 log(1.0/0.2236) = 13 dB.

As is typical with most spectrum analyzers, the vertical axis provides magnitude (volts) or power (watts) information rather than power spectral density information in Watts/Hz. The amplitude and frequency values are easily verified using Cursors. Select "Cursors" (on the main menu bar), "Cursor Type," "Paired," and then place the two cursors on the fundamental and 3rd harmonic. When this is done, the on-screen display provides frequency and amplitude details at the cursor locations. For this example, the amplitudes are -0.824 dBW and –10.29 dBW, which correspond to the square wave's Fourier series rms coefficients $|c_1|={}^{2\sqrt{2}}\!/_{\pi}$ and $|c_3|={}^{2\sqrt{2}}\!/_{3\pi}$.

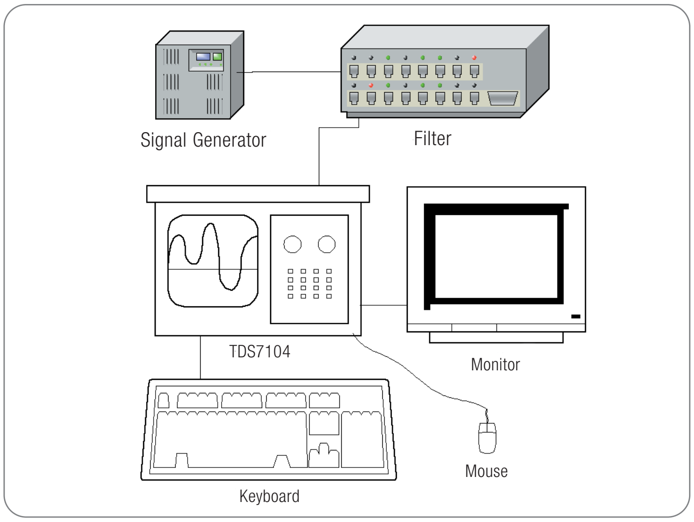
It is also interesting to investigate the effects of different duty cycles. For example, Figure 4 shows that increasing the duty cycle from 50% to 65% causes both odd and even harmonics to become visible. If higher spectral resolution is required, the record length and gate duration can easily be increased. For example, doubling the record length to 5000 reduces the width of the spectral lines by one half. Increasing the record length and gate duration to 25000 samples provides a factor of ten improvement in resolution, as demonstrated in Figure 5.



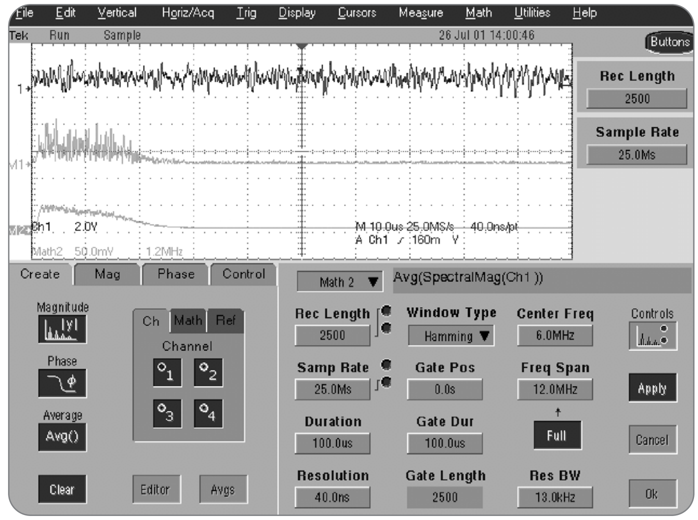
▶ ***Figure 4.*** *Time domain trace and corresponding spectrum for 65% duty cycle square wave.*



▶ ***Figure 5.*** *Spectrum of 65% duty cycle square wave with longer record length and increased resolution.*



▶ ***Figure 6.*** *Setup for measuring transfer function of a filter.*



▶ ***Figure 7.*** *Oscilloscope screen showing the raw signal trace (top), spectrum of single trace (middle) and averaged spectrum (bottom).*

The first example described the analysis of a *deterministic* signal. In many applications the signal of interest is non-deterministic or random. As an example, we next study the output of a linear filter when the input is white noise generated using the setup shown in Figure 6. Simple Fourier techniques, such as the simple periodogram may not yield good results on random data, as the Fourier transform may not even formally exist for this type of signal. An averaged Welch Periodogram will perform better.

The oscilloscope is set up with the following parameters:

- Sample rate f_s = 25 MHz, frequency span = 8 MHz, center frequency = 4 MHz
- Record length = 2500 samples for a duration of 100 µs
- Math 1 displays the magnitude spectrum while Math 2 displays an averaged spectrum (discussed below). Both are displayed using linear scale.

Theoretically, the signal spectrum for this case is given simply by:

$$\begin{aligned}\hat{P}(f_k) &= \hat{P}_N(f_k)\left|H(f_k)\right|^2 \text{, Watts/Hz} \\ &= \left|H(f_k)\right|^2\end{aligned}$$

where $H(f_k)$ is the filter transfer function, and the input noise is assumed to be white with unit spectral density $\hat{P}_N(f_k) = 1$. Since the Tektronix linear display gives an *amplitude* spectrum (i.e., the square root of the power spectrum), the oscilloscope would ideally give the magnitude transfer function of the unknown filter as:

$$\sqrt{\hat{P}(f_k)} = \left|H(f_k)\right|$$

From the spectrum plot in Figure 7, you can see that the linear filter has a band-pass behavior with cut-off frequencies of about 600 kHz and 1.3 MHz. However, the magnitude spectrum (middle trace) is very noisy with only the general shape of a band-pass filter response. This is because the displayed result is based on the simple periodogram spectrum discussed above and thus is very noisy. Notice that the displayed spectrum $D(f_k)$ is related to the simple periodogram by both a scaling factor and a square root:

$$D(f_k) = \sqrt{\frac{f_s}{N}\hat{P}(f_k)} \text{, (Volts)}$$

A smoother estimate can be obtained by averaging the spectrum over M consecutive data segments:

$$\hat{P}(f_k) = \frac{1}{M}\sum_{m=1}^{M}\hat{P}_m(f_k)\text{, (Watts/Hz)}$$

where $\hat{P}_m(f_k)$ is the simple periodogram spectrum of the m-th data segment. Obtaining this exact result is not straightforward; a similar result can be achieved as follows. Using a linear plot scale, compute a smoothed amplitude spectrum by selecting Math 2, clicking Magnitude on Channel 1, and then clicking Avg(). The result is:

$$D(f_k) = \frac{1}{M}\sum_{m=1}^{M}D_m(f_k)\text{. (Volts)}$$

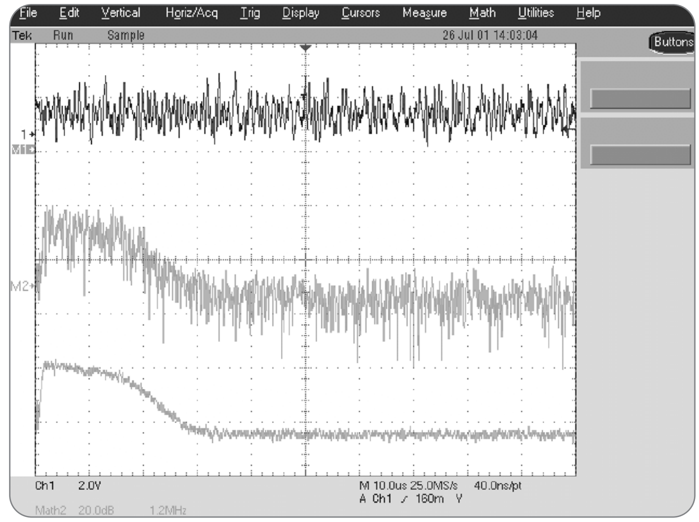
The number of averages, M, dictates how smooth the result is, and can be modified by clicking on the **Avgs** button on the Spectral Setup screen (under the Create tab). This will bring up a screen where the number of averages can be chosen for each of the four Math functions. The result for M = 20 is shown as the bottom trace in Figure 7, and is a significantly improved estimate of $|H(f_k)|$. Due to the differences evident in the above equations, the result is not a rigorous estimate of $|H(f_k)|$, however it does provide a useful characterization of the filter. More accurate estimates using MATLAB are discussed in the next section.

In many cases, it is more useful to use a dB plot scale rather than a linear amplitude scale. You can compute the average spectrum with the dB scale enabled (under the "Mag" tab) such that the resulting displayed spectrum is the average of the scaled log spectra (see Figure 8):

$$\hat{\hat{P}}_{dB}(f_k) = \frac{1}{M}\sum_{m=1}^{M}10\log\left(F_s\ \hat{P}_m(f_k)\right)\text{ (dB Watts)}$$

This is quite different from the power spectral density estimate discussed in section 1:

$$\hat{P}_{dB}(f_k) = 10\log\left[\frac{1}{M}\sum_{m=1}^{M}\hat{P}_m(f_k)\right]\text{(dB Watts/Hz)}$$



▶ ***Figure 8.*** *Oscilloscope screen showing the raw signal trace (top), log spectrum of single trace (middle) and averaged log spectrum (bottom).*

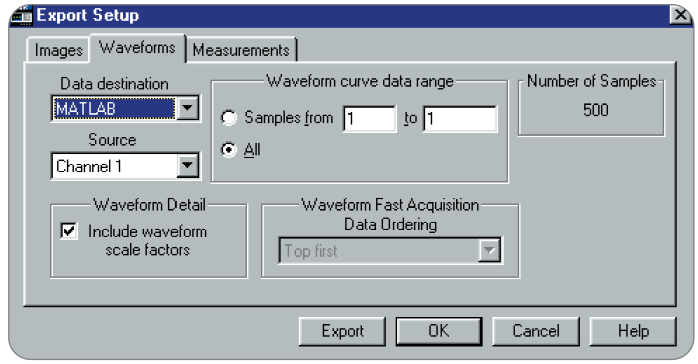
The result in Figure 8 is much smoother, and provides a more visually useful display than the non-averaged spectrum. Since the order of the summation and the logarithm operations are swapped, there is no clear relationship to the true spectral density.

In applications where properly scaled and averaged power spectral density estimates are required, the computations are readily computed by exporting the data trace to MATLAB for post processing. This powerful capability of the TDS7000 is the focus of the next section.

Spectrum Measurement Using MATLAB

MATLAB from The Mathworks Inc. is a very extensive tool for mathematical processing and analysis of data. Combine MATLAB with Mathworks' Signal Processing Toolbox and you have an extremely powerful tool for processing signals off line. In addition to the Welch periodogram, MATLAB includes at least eight alternative methods for estimating a power spectrum. These methods are briefly mentioned in this section, and further information is available through MATLAB's online help. A brief discussion of other signal processing capabilities within MATLAB and the Signal Processing Toolbox is presented here. Although MATLAB is a very extensive program, many operations can be performed with minimal background experience.

This section describes the process of exporting data from a TDS7000 Series oscilloscope to the MATLAB environment. Several examples of computing Welch power spectrum estimates using MATLAB and its Signal Processing Toolbox are provided. These examples include another look at the bandpass filter from the previous section as well as an example of radiated emissions from an Adjustable Speed Drive (ASD).

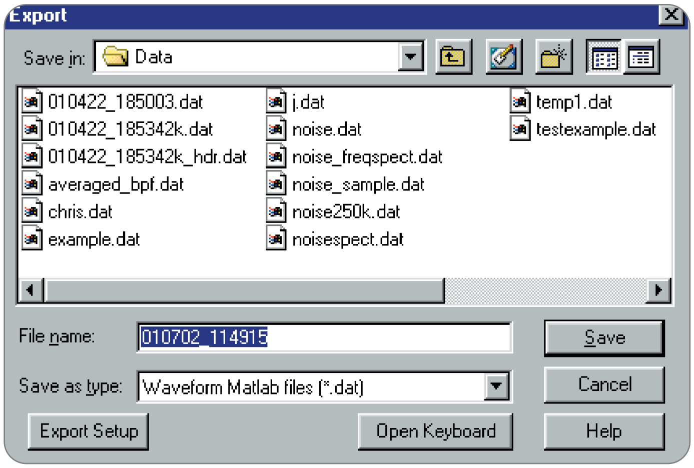


► ***Figure 9.*** *The Export setup window.*

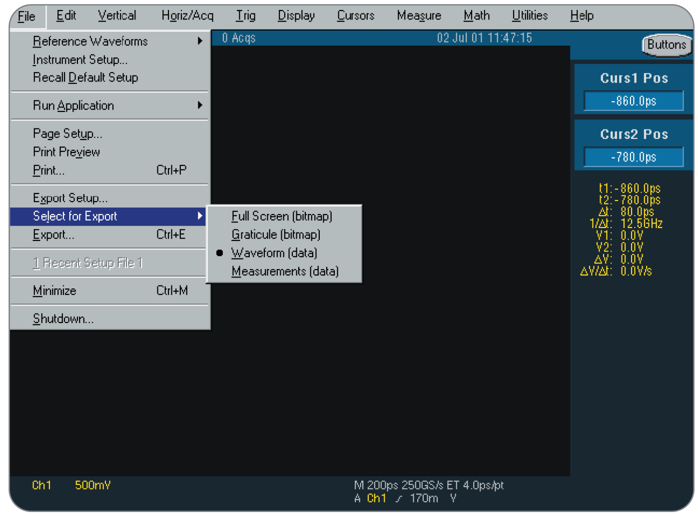
Exporting Data to MATLAB

Assuming that the oscilloscope is displaying the data to be analyzed, it can be exported to a MATLAB-compatible file for further processing using the following steps:

- Under the **File** menu button click on the **Export Setup** button. A window resembling that shown in Figure 9 will be displayed.
- Once inside the **Export Setup** window click the **Waveforms** tab.
- Change the Data Destination to **MATLAB**.
- Choose the desired Source (**Channel 1-4**, **Math 1-4**, **Reference 1-4**).
- There are also two other options. Waveform Detail can have **included waveform scale factors** and you can also select the data range for the waveform (for this application the whole sample will be exported). This data range can also be easily manipulated with MATLAB.
- When all the proper settings have been selected the **OK** button should be clicked.
- Next click the **File** button on the main menu bar. Click the button **Select for Export**, as shown in Figure 10 and select **Waveform** (data).
- When the proper settings have been selected, click the **File** button and then click the **Export** button. An Export save window, as shown in Figure 11 will appear.
- Select the directory, filename, and type of file. For this application note the files will be saved as **Waveform MATLAB files [*.dat]**.
- Click the **Save** button.



▶ ***Figure 10.*** *Waveform (data) must be selected from the "Select for Export" button.*



▶ ***Figure 11.*** *The Export Save window.*

Now that the data has been saved onto the hard disk, it can be opened in MATLAB, analyzed, and manipulated as desired. Assume that the file is saved as **signal.dat** in the \data directory. After starting up MATLAB, change the directory and load the data as shown:

cd \data

load signal.dat

The sampled data is now loaded into a MATLAB variable called "signal," which can be plotted or processed. For example, the command

plot(signal(5000:20000))

creates a plot of data samples 5000 through 20000.

```
load signal.dat      % loads the data into the MATLAB working environment
x=signal;     % places the signal data into the variable x
clear signal %clears memory space taken by "signal"
fs=25e6;   % the sampling rate in samples/s
Ts=1/fs;  % the sampling period
N=length(x);    %  Record length of data
L=2500;   % length of each segment
M=fix(N/L);     % the number of segments to be averaged
K=3000;  % number of points in the frequency domain
w=hamming(L);    % w[n], the Hamming window of length L
U=sum(w.^2)/L;     % the windows coherent gain
[P,f]=psd(x,K,fs,w,0);  % welch periodogram with no overlap
P=((2*Ts)/(L*U))*P;   % apply the appropriate scaling units for W/Hz
P=10*log10(P);      % convert to dB
f=f*1e-6;    %convert horizontal axis values to MHz
figure(1)          %generate a Figure window
plot(f,P)          %plot the estimated power spectrum P as fcn of freq f
xlabel('f (Hz)')    % label the frequency axis
ylabel('P (dB)')    % label the amplitude axis
title('Power Spectral Estimate')
```

▶ ***Figure 12.*** *MATLAB script file to compute and plot the Welch periodogram.*

Spectrum Analysis

The MATLAB Signal Processing Toolbox contains a number of functions for computing estimates of the power spectrum. This example focuses on the *psd* (power spectral density) function, which computes a Welch averaged periodogram to within a scale constant. The command line for executing the "psd" function is

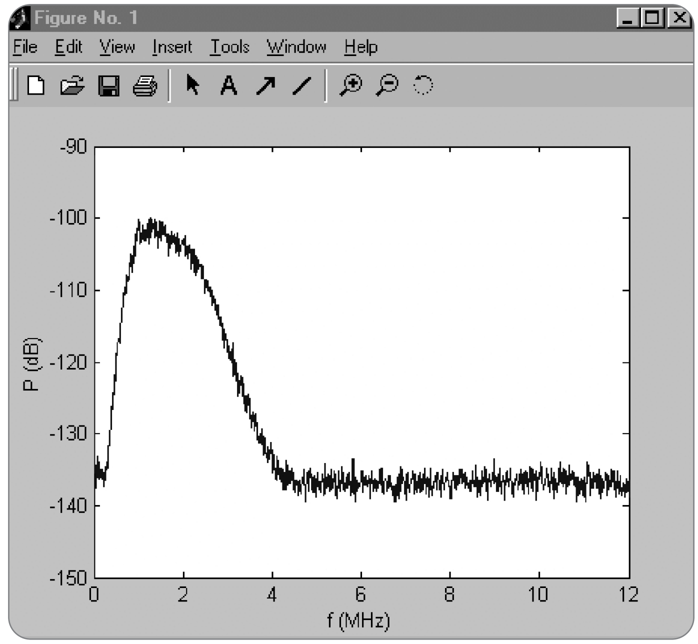
$$[P,f] = psd(x,K,fs,w,overlap);$$

Input variables are in parentheses on the right while output variables are within brackets on the left. The variable x is a vector containing the sampled signal of length N. The variable K is the number of points to compute in the frequency domain and fs is the sampling rate. The variable w is a vector containing the window values of length L and *overlap* is the number of points by which the segments overlap. The variable P is a vector of values containing the power spectrum estimate at the corresponding frequencies in the frequency vector f.

A MATLAB script program file that uses the psd function and plots the result is shown in Figure 12. A MATLAB script file is a program that executes one command at a time just as if you typed in each command separately at the MATLAB command prompt. Note that text preceded by a % sign is considered to be a comment. To execute the script file, just type its name at the MATLAB prompt. If the file is named *pspec.m*, the script is executed by typing *pspec* at the prompt.

The first few lines of the code load the saved data file into MATLAB and moves the data into the variable x (this step is optional). The next few lines of the code create the other input parameters such as sampling rate (f_s), and the window function (w). MATLAB has a number of different window functions available including the Hamming window as shown. The command w=hamming (L) creates a vector of length L corresponding to the Hamming window function. Other window functions included in MATLAB, which are also on the TDS7000 Series oscilloscope include hanning, boxcar (rectangular), and Kaiser. MATLAB also has a number of other windows. In the code shown, after the window is defined, the coherent gain factor (U) is calculated and used to properly normalize the estimate.

The next few lines use the "psd" command to compute the estimated Welch averaged periodogram. The appropriate scale factors are used to convert the units to V^2/Hz, and then the data is converted to dB scale. The last few lines of the script file are the commands to plot the estimated power spectrum as a function of frequency and to label the axis.



▶ ***Figure 13.*** *Welch periodogram of filtered white noise using MATLAB.*

As an example of applying MATLAB spectral analysis, we look at the example of the filtered white noise discussed earlier. The data is first transferred to MATLAB as described above. The sampling rate is set to 25 MS/s and the total record length is set to 50,000 samples. The data is then stored as **signal.dat** and loaded into MATLAB for processing. The MATLAB script file shown in Figure 12 is used to compute and plot the estimated power spectrum. Here the segment length is set to L = 2500 samples to make it easy to compare with the oscilloscope results presented in Figures 7 and 8 which used gate duration of 2500 samples. No overlapping of the segments is used in this example so the number of averages is M = 20. The resulting estimate of the power spectrum is shown in Figure 13. The shape of this power spectrum estimate is similar to the one shown in Figure 8.

```
p=0.99;                      % select a 99% confidence interval
[P,Pc,f]=psd(x,K,fs,w,0,p);  % welch periodogram with no overlap
Pc=((2*Ts)/(L*U))*Pc; % apply the appropriate scaling units of V^2/Hz
Pc=10*log10(Pc);        % convert to dB
f=f*1e-6;            %convert x-axis values to MHz
figure(2)                  % create a second Figure window
plot(f,Pc(:,1),f,Pc(:,2))  %plot the upper and lower 99% bounds
axis([0 12 -90 -30])
xlabel('f (MHz)')           % label the frequency axis
ylabel('P (dB)')          % label the amplitude axis
title('99% Confidence Interval for Power Spectral Estimate')
```

▶ ***Figure 14.*** *MATLAB script to compute and plot confidence intervals.*

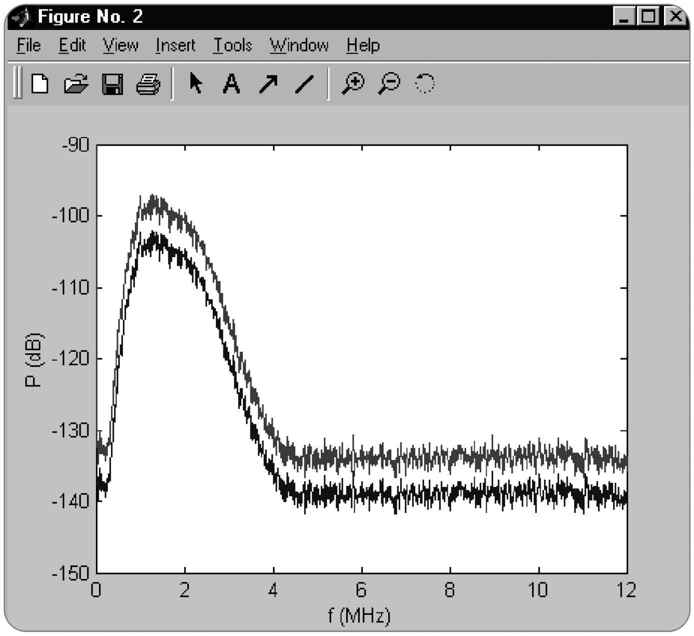
A powerful use of MATLAB in off-line processing of the data is the flexibility to reprocess the same data set using different windows, different segment lengths, or different amounts of overlap to get the desired level of tradeoff between frequency resolution and variability in the spectral estimate. In the MATLAB environment a number of preprocessing steps can be performed on the data before estimating the spectrum. For example, the data could first be filtered, decimated or detrended.

Two additional input parameters can be provided as arguments to the psd function. One is for detrending each segment of data before computing the power spectral density estimate. The other is an input parameter to compute an approximate confidence interval for your power spectral density estimate. The modified psd command is as follows:

[P,Pc,f] = psd(x,K,fs,w,overlap,p,dflag).

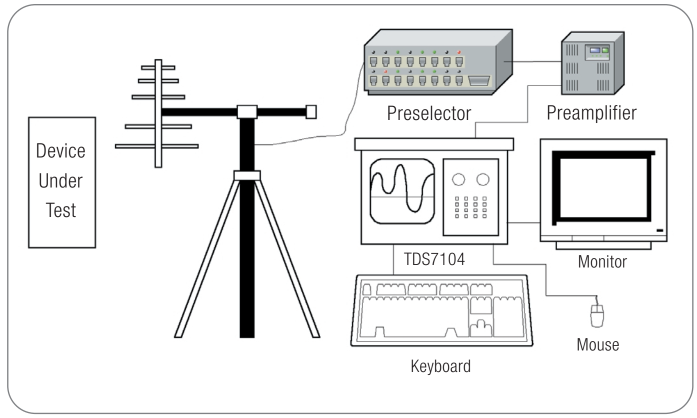
The input variable *dflag* can be set to **none**, **mean**, or **linear**, which sets the detrending mode for the prewindowed data segments. This option is useful if there is a DC offset or an underlying linear trend in the data. The input parameter *p* is a number between 0 and 1. If for example *p=0.99* then MATLAB will compute the 99% confidence interval for the power spectrum estimate. The output variable *Pc* includes two data columns where the first column is the lower boundary and the second column is the upper boundary of the 99% percent confidence interval. It should be noted that the confidence interval estimate is approximate and only valid when zero overlap is used in the data segments.

Figure 14 shows a MATLAB script file that is executed after the script file in Figure 12 to compute and plot the estimated power 99% confidence intervals. The result is shown in Figure 15.



▶ ***Figure 15.** Upper and lower 99% confidence bounds on Welch periodogram.*

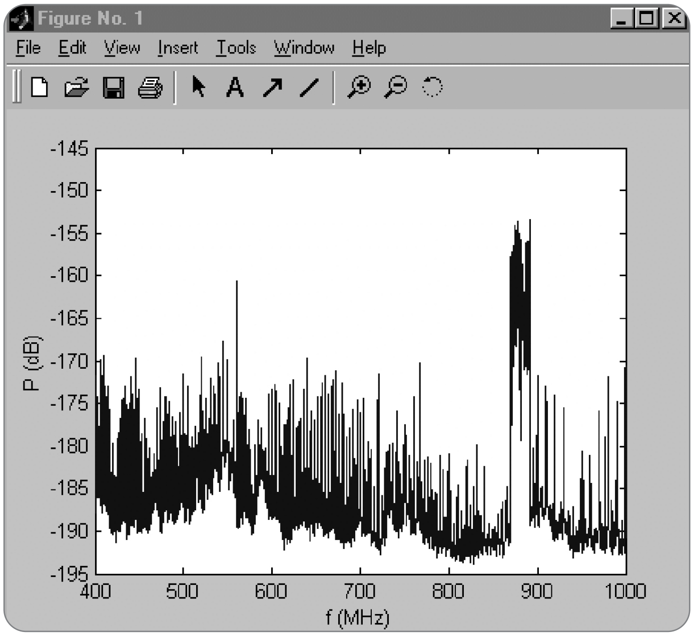
The final example uses the extensive bandwidth and capabilities of the TDS7000 Series to measure the radiated emissions from an adjustable speed drive (ASD). ASDs are power conversion devices that convert 60 Hz to other frequencies, usually using pulse width modulation. ASDs have widespread application in controlling motors for elevators, air handling units, etc. One of the undesired side effects of ASDs is radiated emissions.



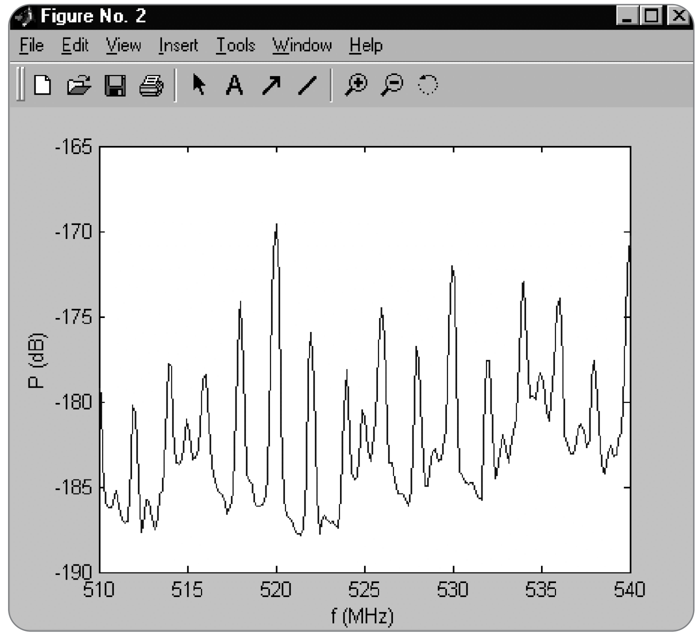
▶ ***Figure 16.** Setup for measuring EMI from adjustable speed drive.*

In this experiment a log-periodic antenna is used to detect the emissions. Connect the antenna to the oscilloscope's 50 ohm input through a preamplifier and preselector (see Figure 16). The ASD is controlling a motor and the antenna is at a distance of 3 meters from the ASD. The sampling rate is set to 5 GS/s. A record length of 750 KS/s is acquired and transferred to MATLAB for processing. More resolution can be obtained by transferring larger data segments from the oscilloscope. Here is a case where the additional memory options for the TDS7000 Series can be useful for achieving greater record length and thus better resolution in the frequency domain.

The previous MATLAB code is used to analyze the data. In this case, L = 20,000 samples and K = 32,768 (2^{16}). The overlap is set to 50% (10,000 samples) which results in a value of M = 74 averages. A Hamming window is used. Figure 17 shows the estimated power spectrum from 400 MHz to 1 GHz. Note, the clearly visable cell phone activity near 900 MHz. Using the "zoom" feature in MATLAB, Figure 18 shows the range from 510 to 540 MHz. This Figure clearly shows harmonically related interference coming from the ASD spaced every 2 MHz.



▶ ***Figure 17.*** *Welch periodogram for an ASDs radiated emissions between 400 MHz and 1 GHz.*



▶ ***Figure 18.*** *Welch periodogram, zoomed in on the radiated emissions between 510 MHz and 540 MHz.*

Conclusion

Many lab-quality oscilloscopes include spectral analysis functions in addition to their time-domain measurement features. While this capability has been available for some time now, recent advancements have brought some of these spectral analysis features to parity with dedicated spectrum analyzers in many respects.

A high-performance oscilloscope equipped with spectral analysis capability, like the TDS7000 Series DPO, is a multi-faceted solution for design evaluation and debug work, where frequency-domain measurements can reveal system behaviors that are not apparent in time-domain waveform displays.

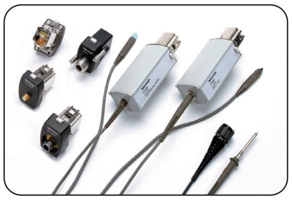
Written by:

John W. Pierre, Ph.D.
Associate Professor, Electrical Engineering
University of Wyoming

Robert Kubichek, Ph.D.
Associate Professor, Electrical Engineering
University of Wyoming

TDS7000 Series Digital Phosphor Oscilloscopes

The TDS7000 Series oscilloscopes, with bandwidth from 500 MHz to 4 GHz and up to 20 GS/s real-time sample rate, are high-performance real-time oscilloscopes for verification, debug and characterization of sophisticated electronic designs.



P7330 Probes

The P7330 provides high bandwidth, low circuit loading, low noise differential probing for time or frequency domain measurements on high bandwidth signals such as Fibre Channel. The P7240 single-ended probe with TekConnect interface is Tektronix' highest performance active prove, providing low capacitive loading and high bandwidth.

AWG710

The AWG710 combines world-class signal fidelity with ultra high-speed mixed signal simulation, a powerful sequencing capability and graphical user interface with flexible waveform editor, to solve the toughest measurement challenges in the disk drive, communications and semiconductor design/test industries.

Contact Tektronix:

ASEAN Countries (65) 356-3900

Austria +43 2236 8092 262

Central Europe & Greece +43 2236 8092 301

Belgium +32 (2) 715 89 70

Brazil & South America 55 (11) 3741-8360

Canada 1 (800) 661-5625

Denmark +45 44 850 700

Finland +358 (9) 4783 400

France & North Africa +33 (0) 1 69 86 80 34

Germany +49 (221) 94 77 400

Hong Kong (852) 2585-6688

India (91) 80-2275577

Italy +39 (02) 25086 1

Japan (Sony/Tektronix Corporation) 81 (3) 3448-3111

Mexico, Central America & Caribbean 52 (5) 666-6333

The Netherlands +31 (0) 23 569 5555

Norway +47 22 07 07 00

People's Republic of China 86 (10) 6235 1230

Poland +48 (0) 22 521 53 40

Republic of Korea 82 (2) 528-5299

Russia, CIS & The Baltics +358 (9) 4783 400

South Africa +27 11 254 8360

Spain +34 (91) 372 6055

Sweden +46 8 477 6503/4

Taiwan 886 (2) 2722-9622

United Kingdom & Eire +44 (0) 1344 392400

USA 1 (800) 426-2200

For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit **www.tektronix.com**

01/02 HMH/XBS 55-15429-0

Tektronix
Enabling Innovation