

Arbitrary FIR Filter Theory, Design, and Application



Oscilloscopes have incorporated the ability for the user to apply their own custom designed FIR filters to math waveforms. This paper provides an in depth discussion of the theory of FIR, finite impulse response, filters using DSP. It also describes some methods for designing these filters. In addition there is a discussion of the FIR filter library provided for use with the Tektronix arbitrary filter used in the oscilloscope math subsection. This paper deals only with FIR filter types.

Section 1: Introduction

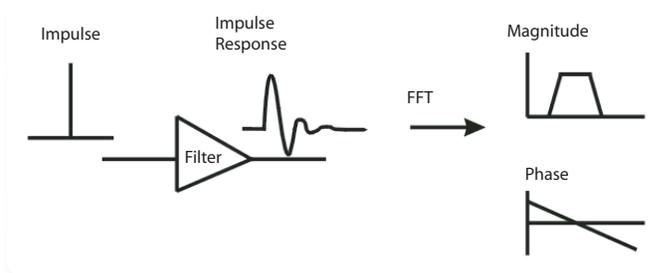
In terms of signal processing, a filter may be thought of as a mathematical procedure that modifies the shape of a waveform. Thus given a waveform of any shape a filter could be designed that would transform it into any desired shape within the context of some basic rules, assumptions, and limitations. In that broad

sense one might say that any system that processes a signal could be thought of as a filter. For example, an oscilloscope channel operates as a lowpass filter where its 3 dB down point is referred to as its bandwidth. If AC coupling is used then the oscilloscope operates as a bandpass filter.

Before the advent of computers and digital sampling, engineers dealt primarily with analog filters implemented in electrical circuits. These filters used resistors, capacitors, and inductors to perform an “analog computation” known as a convolution. With the invention of digital computers and A/D converters the convolution process can be performed by a series of multiplications and additions on binary data samples that represent a signal.

Digital filters have some significant advantages over analog filters. For example, the tolerance values of analog filter circuit components are large enough such that high order filters are difficult or impossible to implement. With digital filters such high order filters are easily realized. In addition, analog component values can change with age or temperature affecting the response of the filter. Digital filters do not have that problem. Another major advantage of digital filters is the ability to reprogram them by changing the coefficients. This greatly simplifies the implementation of adaptive filters.

Generally speaking, digital filters fall into two major categories, FIR “finite impulse response”, and IIR “infinite impulse response”. The FIR filter has only zeros and no poles in its transfer function. Thus it is always stable and cannot oscillate. Therefore, the impulse response of an FIR filter has finite length. Also, the FIR filter may be specified with an exactly linear phase response. An IIR filter has both zeros and poles in its transfer function and can be unstable. Its impulse response theoretically lasts to infinity. In other words, it is implemented with a feedback loop. An IIR filter cannot obtain a true linear phase response. However, it can approximate linear phase over regions of interest. This paper will only deal with filters of type FIR. For a more in depth description of digital filter theory, see reference list on page 15.



► **Figure 1.** Basic Filter Conceptual diagram.

Section 2: FIR Filter Coefficients

An FIR filter consists of an array of data samples, $h(m)$, that represent the sampled impulse response of the filter where m is the index into the array. Refer to Figure 1. It is well known that an impulse consists of cosine functions of all frequencies of equal magnitude with phase equal to zero at the time position of the impulse, i.e. each cosine wave of a different frequency has a peak at the position of the impulse. This position is referred to as the zero phase reference point. Thus, if an impulse is fed into a filter, its impulse response will fully define the characteristics of the filter in the time domain. Computing an FFT of the filter impulse response will provide the phase and magnitude response of the filter in the frequency domain.

An FIR filter is obtained by specifying its coefficients.

The coefficients are samples of the filters' impulse response. There are numerous methods by which the impulse response may be obtained. Generally the filter characteristics are specified in the frequency domain and the impulse response is obtained by computing an IFFT. The impulse response is then sampled to obtain the filter coefficients at the desired sample rate.

Remez Exchange Method: Perhaps the most well known method for FIR filter design uses the Remez exchange algorithm. This algorithm is described in “Theory and Application of Digital Signal Processing” (see reference list on page 15) where a FORTRAN program listing is provided. This program allows the user to specify one of three types of linear phase FIR filters such as 1) Multiband, 2) Hilbert Transform, and 3) Differentiator. Matlab™ has a filter design package that uses the Remez algorithm providing a modern Windows™ user interface.

Analog Prototype: There are also methods for transforming a known analog transfer function into a digital filter transfer function using a bilinear transform. This is needed because the representation of the complex $j\omega$ axis in the analog s -plane maps onto a unit circle in the digital z -plane domain. The bilinear transform is used to pre-warp the filter transfer function when converting it from the s -plane to the z -plane to obtain an equivalent digital filter transfer function.

A Direct Frequency Sampling Method Used By the

Author: The Remez algorithm method and analog prototype methods described above have significant limitations in regards to controlling both the phase response and the magnitude response of the desired filter. It can be difficult to obtain a desired response with these methods. In Section 8: “Arbitrary Frequency Sampling Filter Design Method,” a procedure is given to create a filter with arbitrary magnitude and phase response as specified by the designer.

Section 3: Convolution

Once the filter has been designed and the filter coefficients have been obtained the application of the filter to the input data is performed by the process of convolution. This is described in equations (1), (2), and (3) in a form that is easily implemented in

Mathcad™ or Matlab™.

$$m := 0 \dots (M - 1) \tag{1}$$

$$n := \text{floor}(M/2) \dots (N - 1 - \text{floor}(M/2)) \tag{2}$$

$$y(n) := \sum_m x(n - m + \text{floor}(M/2)) \cdot h(m) \tag{3}$$

$$p = 0 \dots \text{floor}(M/2) - 1 \tag{4}$$

$$x(p) = y(p) \tag{5}$$

$$y(N - p - 1) = x(N - p - 1) \tag{6}$$

Where M is the number of filter coefficients and must be an odd number for the indexing scheme described in equation (2). Even length filters can also be implemented but the indexing equation (2) must be modified slightly. The value of N is the length of the input data array specified by x . The filter array is h . As can be seen by equation (1), (2), and (3) the process of convolution involves reversing the time order of the impulse response of the filter, h , and multiplying it point by point with corresponding samples of x starting at index position M . The results of each multiply are added together to obtain one filtered output point into the array, y . The value of n is then incremented by one and the multiplication and summation procedure is repeated to obtain the next filtered data sample in y .

Notice that as specified by equation (3) the first $\text{floor}(M/2)$ points in the $y()$ array are not defined. Also, the last $\text{floor}(M/2)$ points at the end of the array are not defined. However, the remainder of the points in the array are filtered. With this representation the filter is fully charged and there is no transient start up distortion from the filter in the output data array. All points of the final array, $y(n)$, described by equation (6) are valid.

The oscilloscope filtered math waveform does not have extra points before or after the ends of the waveform record. Therefore, to avoid filter startup transient distortion, the undefined points from equation (3) are replaced with ETnull, “not a number”. These display as blank on screen. In an exported ASCII file these points appear as spaces.

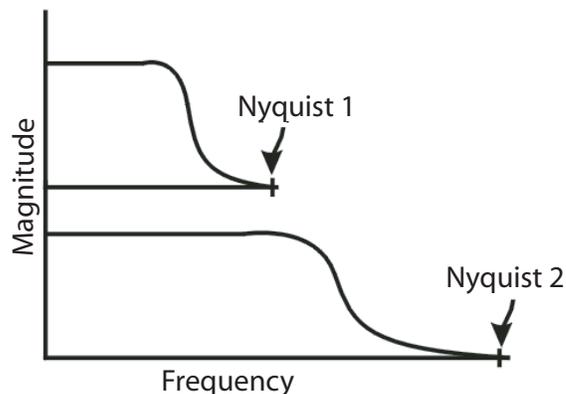
Section 4: Filter Coefficients and Frequency Response With Respect to Sample Rate

The frequency response of any digital filter is normalized with respect to the sample rate, f_s .

The filter will have its magnitude and phase response defined within the frequency range of 0 to $0.5 f_s$ where the value of $0.5 f_s$ is called the Nyquist point. The key filter parameters may be specified using normalized values. For example, a lowpass filter could be specified to have a normalized bandwidth of 0.3. That means that the cutoff frequency is equal to $0.3 f_s$. This filter will have this cutoff frequency no matter what sample rate value is chosen.

For example, consider a lowpass filter that has a pass-band from 0 to $0.25 f_s$. Now assume that the filter coefficients have been determined and that they are applied to data that has some specified sample rate. Regardless of what the sample rate of the data is, the filter will exhibit the same magnitude and phase characteristics scaled to within a range of 0 to Nyquist. Thus if this filter is applied at a sample rate of 1 Hz, the bandwidth would be 0.25 Hz. If the same filter coefficients are applied to data that has a sample rate of 1 GHz, then the filter will have a bandwidth of 250 MHz. A graphic representation of this affect is shown in Figure 2.

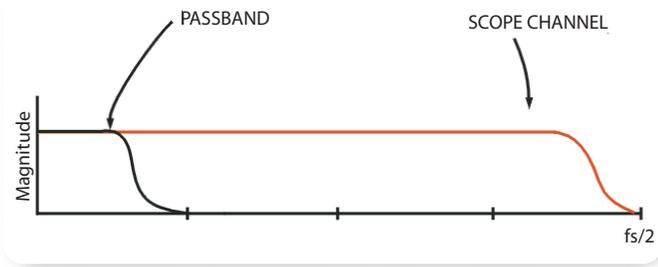
Having a set of normalized filters for general use is valuable. For example, new Tektronix oscilloscopes have such a library of filters stored in disk files. There are highpass, lowpass, bandpass, bandreject, and other types of filters in this library. Refer to Section 10: "The Math Arbitrary Filter Library".



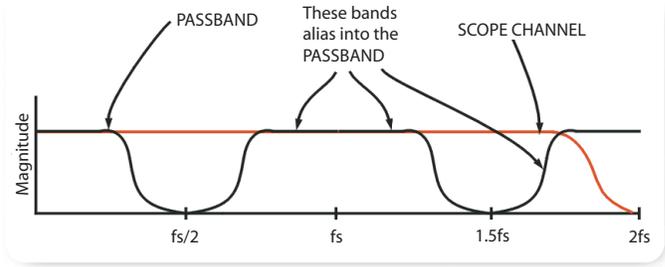
► **Figure 2.** Example of normalized frequency response scaling with same filter coefficients but different sample rates.

Having a fixed set of filter coefficients results in the time duration of the impulse response to vary with different sample rates. Thus the frequency domain characteristics such as bandwidth, will vary accordingly. The number of filter coefficients stays constant for this case.

If the user requires a fixed frequency response from the filter regardless of what the sample rate is then a different set of filter coefficients is required for each sample rate. This means that the time interval covered by the impulse response will be constant. This results in fewer filter coefficients for lower sample rates and more coefficients for higher sample rates. Thus the number of filter coefficients will vary with sample rate for this case. Obviously the sample rate must be high enough such that the desired frequency response fits within the range of Nyquist. For this case the user should design the filter for the highest sample rate since



▶ **Figure 3.** An example of the desired response from a band limit filter in an oscilloscope operating at base sample rate. Filter response is shown in black and scope channel response is shown in red. No aliasing occurs in this example.



▶ **Figure 4.** Example of an aliased Filter Response when the bandwidth limit filter sample rate is too low. Black is digital filter response and red is the analog oscilloscope channel response. The resulting filter does not provide the desired bandwidth limit function.

this will require the most coefficients. Then use interpolation and decimation techniques to resample the filter coefficients for each sample rate the filter shall operate at. The various sets of coefficients may then be incorporated into a single disk file as described in Section 11: “FIR Filter File Format”. When operating at different sample rates in the oscilloscope, the possibility of signal and noise aliasing must be taken into consideration. In addition, the filter will turn off if a sample rate is set in the scope for which no filter coefficients have been defined for the ArbFilt() math function. For normalized filter files as described in Section 11, the filter would always be on regardless of what sample rate is chosen.

Section 5: Aliasing and the Filter Response

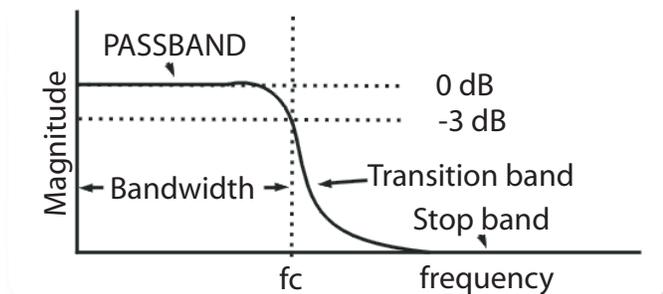
The issue of aliasing must be considered when using FIR filters in general. This is especially true within the math package of an oscilloscope. Any signals that have frequency greater than Nyquist that fall within the analog bandwidth of the oscilloscope will alias within the pass-band(s) and appear to have a lower frequency than the actual frequency.

The best way to avoid aliasing in an oscilloscope is to operate at the base sample rate where the analog bandwidth is less than Nyquist. For this case the analog

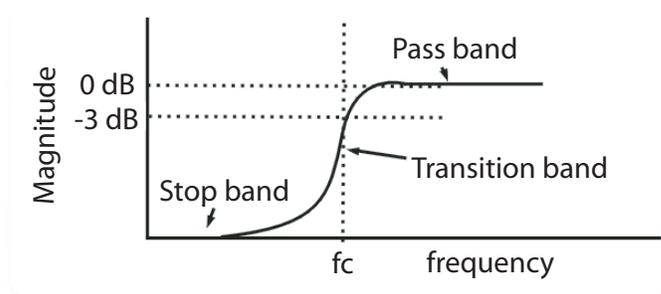
channel of the oscilloscope operates as an anti-alias filter. For example, a DPO7254 has a base sample rate of 10 GS/s and a channel bandwidth of 2 GHz which is less than Nyquist at 5 GHz. This works well as long as the filter bandwidth is not too small compared to the scope bandwidth where the filter would be highly over sampled. The filters are computed with floating point coefficients so they are more tolerant to over-sampling than would be the case for fixed point filters. For this condition the number of coefficients becomes much larger and so does the compute time for the filter convolution.

The graph in Figure 3 shows an example of a low pass filter operating at the base sample rate of the oscilloscope. No aliasing occurs for this example.

An example is shown in Figure 4 where the sample rate, f_s , is lower than the base rate of the oscilloscope and the filter response is aliased. Notice that aliasing of the digital filter results in images that repeat multiple times at integer multiples of the sample rate. In this case the user intended to have a lowpass filter to remove out of band noise; but what they actually get is a lowpass filter with multiple bands of high frequency noise and/or signal aliased into the passband.



► Figure 5. Lowpass filter parameters.



► Figure 6. Highpass filter parameters.

Section 6: Linear Phase

A system that has linear phase as a function of frequency will pass all frequencies with the same time delay. A non-linear phase system will change the phase of different components with respect to each other resulting in waveform distortion. Therefore, it is often desirable to design filters that have linear phase characteristics. However, sometimes a filter is needed that will correct the phase errors in a system so that the filtered result will have a linear phase response. This type of filter will have a non-linear phase characteristic which is complementary to the system phase response to be corrected. An example of this is the bandwidth enhancement filters used in the TDS6154C oscilloscopes. These filters are calibrated during manufacturing and result in a scope channel response that is much flatter and that has a linear phase characteristic.

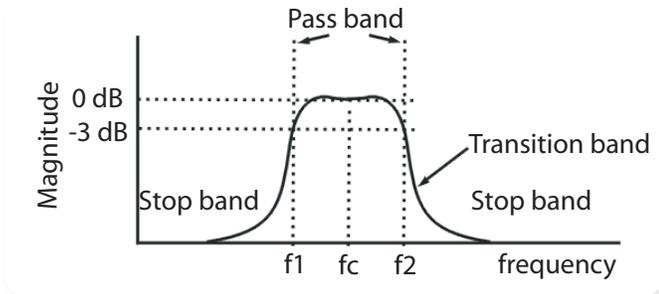
Section 7: Filter Parameters

The following is a list of parameters and terms normally used to describe or specify the characteristics of a filter.

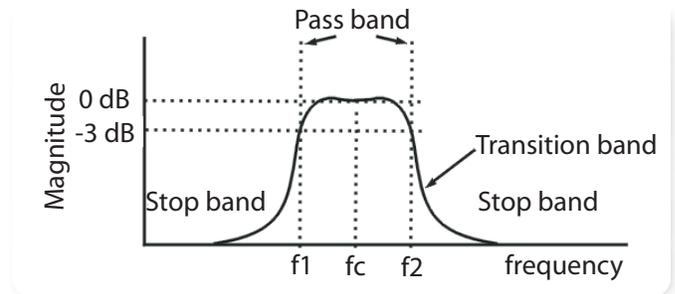
- **Filter Transfer Function Name:** User defined FIR, Bessel Thompson, Butterworth, Chebychev, Gaussian, etc. NOTE: The Bessel Thompson class of filters are

commonly used as the analog bandwidth limit filter in oscilloscope channels. It is also used as the filter response type for optical reference receiver mask testing applications. The reason for its popularity is that it is the closest one can get to realizing an ideal Gaussian filter response which is not physically realizable. This filter type does a reasonably good job of approximating linear phase in the passband using a transfer function with both poles and zeros. NOTE: Several of the filter types listed here originated as analog filter designs incorporating poles and zeros. Therefore to realize them as FIR filters might require using pre-warping and bilinear transform techniques.

- **Filter Type:** Lowpass, highpass, bandpass, bandstop, Hilbert transform, differentiator, multiband, etc.
- **Cutoff Frequency:** Used to specify the bandwidth of highpass, lowpass, bandpass, and band reject filters. Cutoff frequency is the point at which the magnitude has rolled off to -3 dB. Refer to Figures 5, 6, 7, and 8.
- **Bandwidth:** The range of frequencies in the passband of a filter. Cutoff frequencies defining the band edges are at the -3 dB down point on the magnitude response curve.



▶ Figure 7. Bandpass filter parameters.



▶ Figure 8. Bandreject filter parameters.

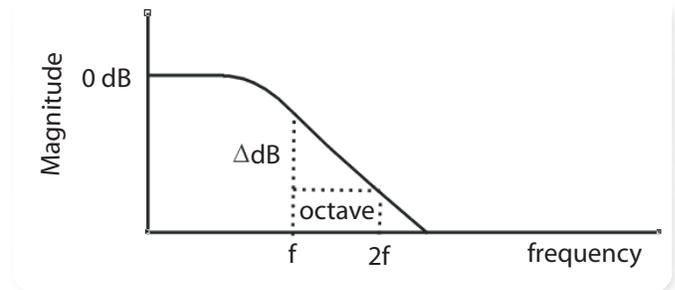
- ▶ **Center frequency:** Used to specify the center frequency of bandpass or band reject filters. Refer to Figures 7 and 8.
- ▶ **Q:** The Q parameter is the ratio of a filters' center frequency to the bandwidth as shown in equation (8).
- ▶ **Rolloff rate:** The slope of the transition band is rolloff rate. It is commonly specified in dB/octave or dB/decade.

Figures 5 and 6 illustrate the various filter parameters and terminology. The filter cutoff frequency is represented by f_c in Figures 5 and 6.

For the case of the bandpass filter shown in Figure 7 f_c is the center frequency of the passband. Also, f_1 is the lower cutoff frequency and f_2 is the upper cutoff frequency and bw is the bandwidth as defined by equation (7). The Q of the filter is defined by equation (8).

$$bw = f_2 - f_1 \quad (7)$$

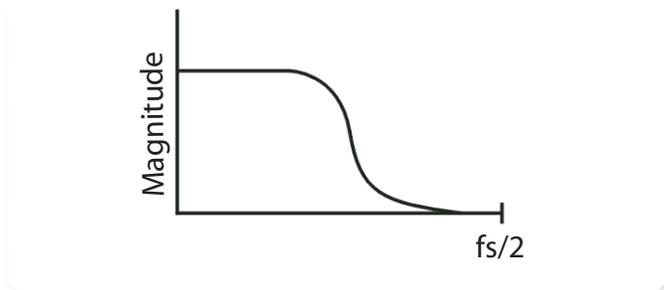
$$Q = \frac{f_c}{bw} \quad (8)$$



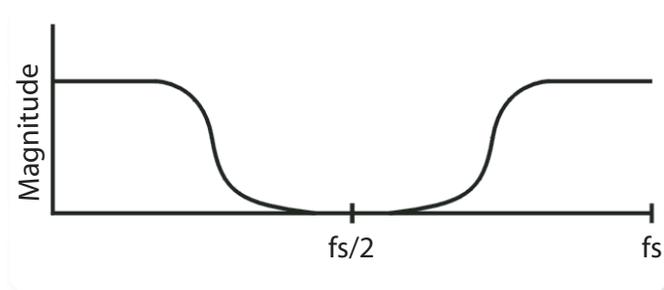
▶ Figure 9. Filter rolloff rate in dB / octave.

Filter Rolloff Rate

The slope of the transition band of a filter is referred to as its' rolloff rate. This is commonly specified in dB per octave or in dB per decade. An octave is an interval where the frequency is doubled. A decade is an interval where the frequency is multiplied by 10. The order of a filter is often associated with the rolloff rate of the filter. For example, a 1st order filter rolls off at 6 dB per octave. A 2nd order filter rolls off at 12 dB per octave. The rolloff rate in dB/octave is equal to the filter order times six. This is only true for typical filter designs where the poles and zeros of the transfer function are positioned to give a typical monotonic slope. It is possible to specify transfer functions for which this rule does not apply.



► **Figure 10.** Define the frequency domain magnitude response.



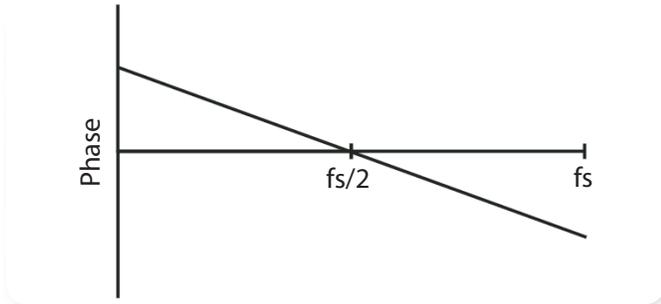
► **Figure 11.** Create the Magnitude portion of the complex conjugate of the data obtained in step 1.

Section 8: Arbitrary Frequency Sampling Filter Design Method

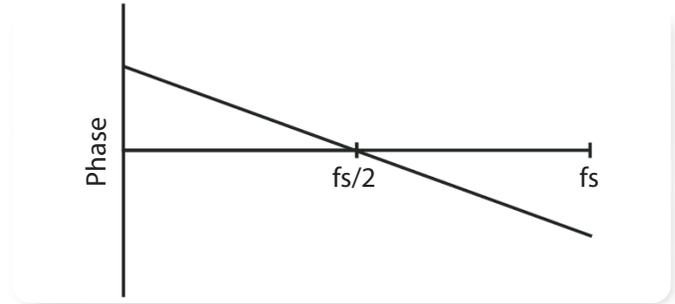
This section describes a design method for arbitrarily specifying both phase and magnitude response for an FIR filter in the frequency domain and then computing the time domain response. Design tools such as Mathcad™ or Matlab™ are ideal environments for implementing these procedures. Refer to “Digital Signal Processing” on the reference list on page 15 for a discussion of frequency sampling filter design method. The general steps are as follows:

- 1. Define the Magnitude response:** Derive a function or other means for specifying the magnitude response of the filter in the range of 0 Hz to the Nyquist point and store in the array, X (See Figure 10).
- 2. Replicate this magnitude data, X, in reverse** order appended to the end of the array, X. See Figure 11. Extreme care must be taken as to the position of

these samples and which index to start on for the beginning and end of the copied data. The total number of points in X will be the length of the IFFT. If the IFFT has an even number of points then the Nyquist point is on one of the samples in the spectrum. If the IFFT has an odd number of points then Nyquist does not occur on a sample but rather between two samples. Failure to take this detail into account will result in errors in the final filter result with the complex part of the IFFT not being equal to zero as it should be. Note: The IFFT of X will be zero if X has a magnitude response symmetrical about Nyquist as shown in Figure 11 and phase response anti-symmetrical about Nyquist as shown in Figure 12. That means the data to the right of Nyquist is the complex conjugate in reverse order of the data to the left of Nyquist where the conjugate of $a + jb$ is equal to $a - jb$.



▶ **Figure 12.** Assign a linear phase function to the data in X. Vertical axis is phase of $X(f)$ where f is frequency.

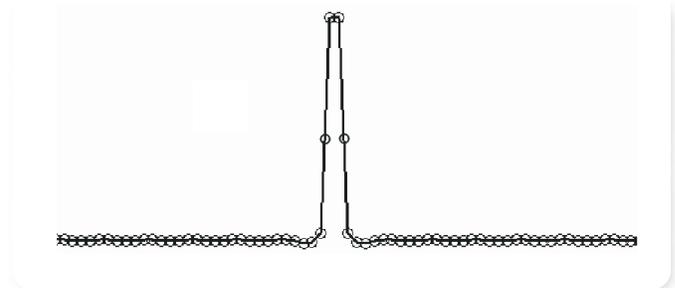


▶ **Figure 13.** Use IFFT to obtain the filter impulse response.

3. Assign Linear Phase: Use rotation of coordinate equation (9) to rotate all of the data points in X. This converts X into an array of complex numbers of the form $a + jb$ where a is the real part and b is the imaginary part and j is the square root of -1 . The equations are as follows:

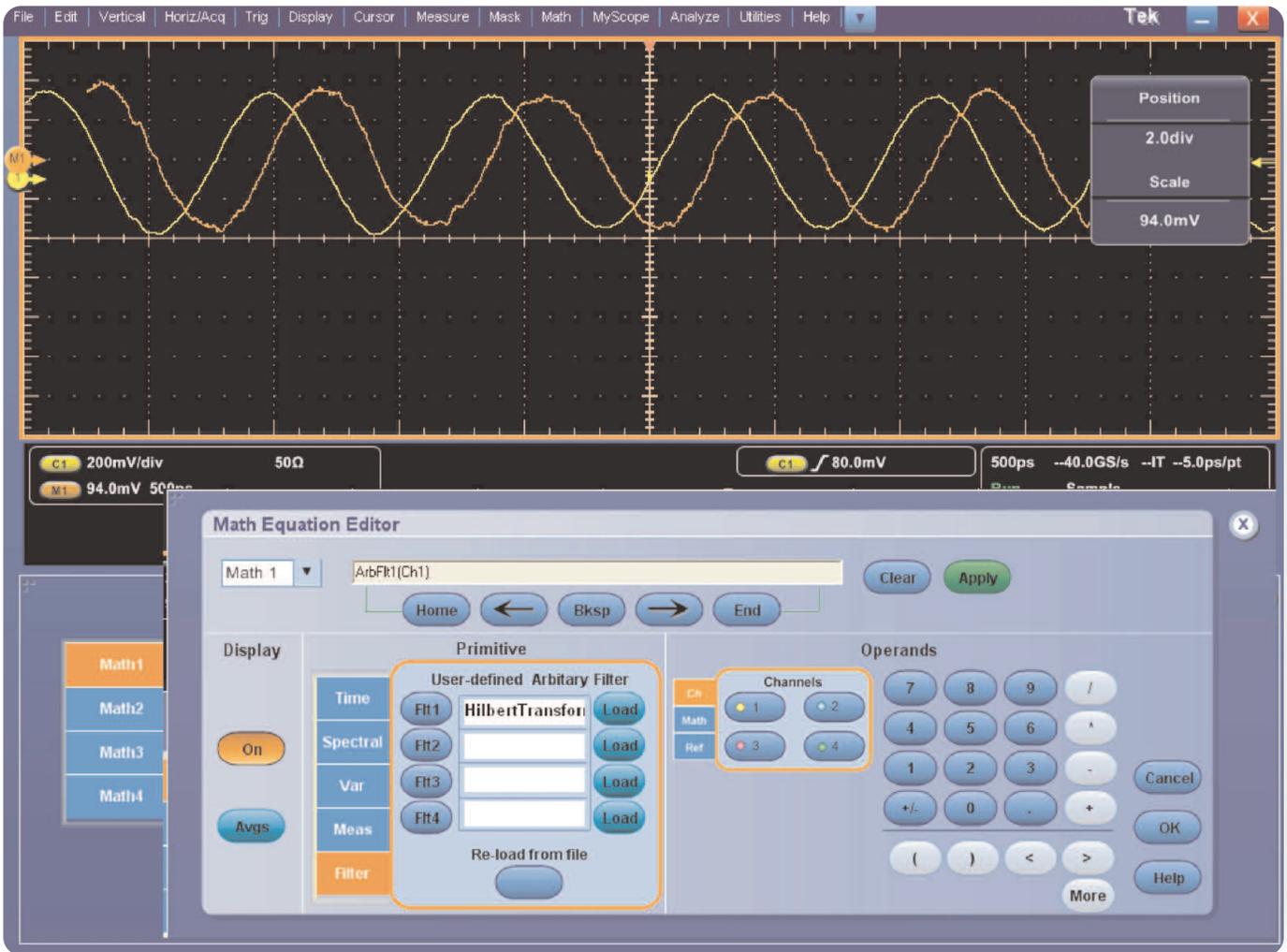
$$\begin{aligned}
 n &= 0 \dots N - 1 \\
 K &= \frac{2 \cdot \pi}{N} \\
 X_n &= X_n \cos(K \cdot n) - j \cdot X_n \sin(K \cdot n)
 \end{aligned}
 \tag{9}$$

Where N is the length of the X array and n is an index variable over the range of the array. The resulting linear phase plot would appear similar to that shown in Figure 12. The group delay is equal to the negative of the slope of the phase response. The steeper the slope of the linear phase response more delay there is through the filter. One may assign no delay to the filter and leave the phase response at zero for all frequency values. For this case equation (9) would not be used.



▶ **Figure 14.** Obtain the final filter coefficients.

- 4. Obtain the time domain impulse response of the filter:** Compute the IFFT of X as shown in Figure 13. If the data indexes were all correct on the above steps to create the frequency domain response and its complex conjugate, then the imaginary part of x will be zero and the real part will be the desired impulse response of the filter.
- 5. Sample the impulse response to obtain the FIR filter coefficients.**



► Figure 15. Screen shot showing `ArbFilt()` function and menu to specify file name for the filter.

Section 9: The DPO Filter Menu

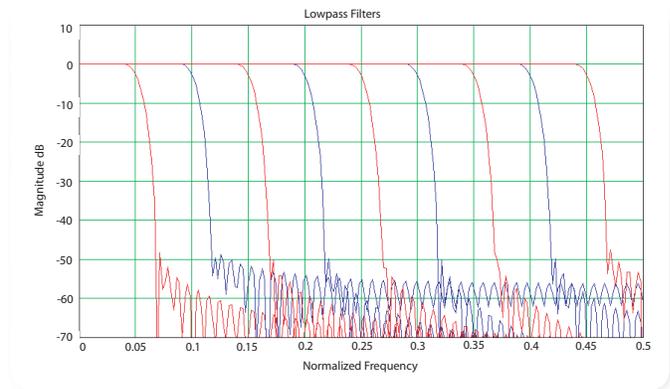
Shown in Figure 15 is a screen shot of the menu used to setup the `ArbFilt1()` math function. This demonstrates how the filter functions may be used in a math expression defined as `math1`.

Section 10: The Math Arbitrary Filter Library

This section describes the library of FIR filters that are available for use under the `arbFilt<x>(<fsource waveform>)` function in the Tektronix oscilloscope waveform math section. The user may find this under the path: `c:\TekScope\Math Arbitrary Filters\<filename>`. The filename of each filter identifies its' type as lowpass, highpass, etc and also identifies its' normalized cutoff frequency or other identifying factors. The precise magnitude characteristics of these filters are shown in the following graphs. These are all linear phase filters.

Lowpass Filters:

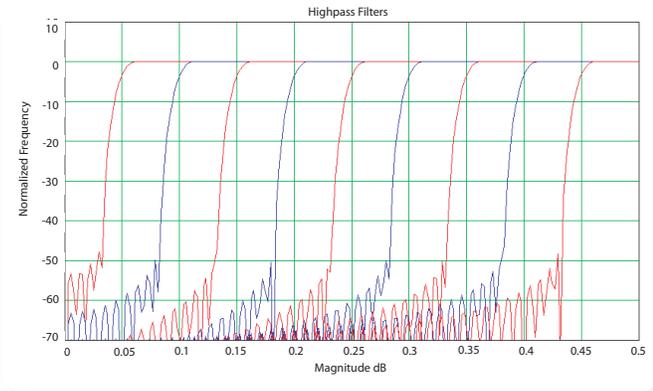
Figures 16 and 17 show the available set of lowpass filters. Their normalized frequency response is shown from 0 to 1/2 the sample rate. These filters will operate at any sample rate with cutoff frequency scaled as shown Figure 16. The filters have normalized cutoff frequencies of 0.05, 0.1, 0.15, 0.20, 0.25, 0.3, 0.35, 0.40, and 0.45. Stop band rejection is typically between -50 and -60 dB.



▶ **Figure 16.** Frequency response of the available lowpass filters.

File Name	Normalized Cutoff Frequency
lowpass_0.05bw.ftt	0.05
lowpass_0.10bw.ftt	0.10
lowpass_0.15bw.ftt	0.15
lowpass_0.20bw.ftt	0.20
lowpass_0.25bw.ftt	0.25
lowpass_0.30bw.ftt	0.30
lowpass_0.35bw.ftt	0.35
lowpass_0.40bw.ftt	0.40
lowpass_0.45bw.ftt	0.45

▶ **Figure 17.** List of lowpass filters available in the library.



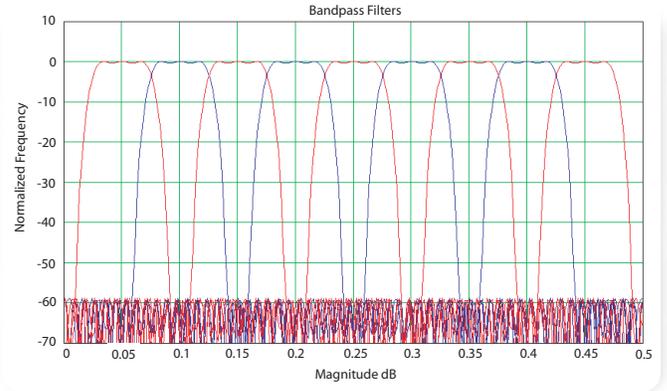
▶ **Figure 18.** Frequency response of the available highpass filters.

File Name	Normalized Cutoff Frequency
highpass_0.05bw.fit	0.05
highpass_0.10bw.fit	0.10
highpass_0.15bw.fit	0.15
highpass_0.20bw.fit	0.20
highpass_0.25bw.fit	0.25
highpass_0.30bw.fit	0.30
highpass_0.35bw.fit	0.35
highpass_0.40bw.fit	0.40
highpass_0.45bw.fit	0.45

▶ **Figure 19.** List of highpass filters available in the library.

Highpass Filters:

Figures 18 and 19 show the available set of highpass filters. Their normalized frequency response is shown from 0 to 1/2 the sample rate. These filters will operate at any sample rate with cutoff frequency scaled as shown below on the graphs. The filters have normalized cutoff frequencies of 0.05, 0.1, 0.15, 0.20, 0.25, 0.3, 0.35, 0.40, and 0.45. Stop band rejection is typically between -50 and -60 dB.



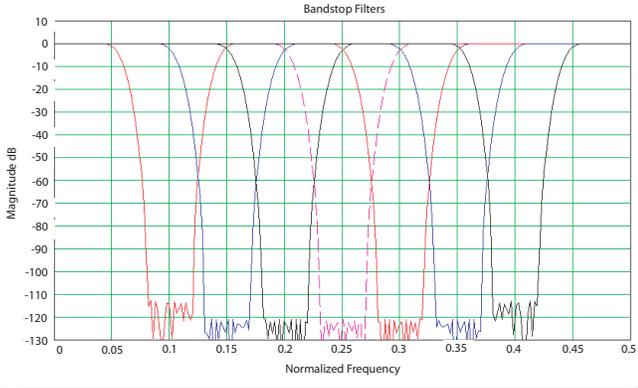
▶ **Figure 20.** Frequency response of the available bandpass filters.

File Name	Normalized Bandwidth	Normalized Center Frequency
bandpass_0.05bw_0.05center.fit	0.05	0.05
bandpass_0.05bw_0.10center.fit	0.05	0.10
bandpass_0.05bw_0.15center.fit	0.05	0.15
bandpass_0.05bw_0.20center.fit	0.05	0.20
bandpass_0.05bw_0.25center.fit	0.05	0.25
bandpass_0.05bw_0.30center.fit	0.05	0.30
bandpass_0.05bw_0.35center.fit	0.05	0.35
bandpass_0.05bw_0.40center.fit	0.05	0.40
bandpass_0.05bw_0.45center.fit	0.05	0.45

▶ **Figure 21.** Table of available normalized bandpass filters.

Bandpass Filters:

Each filter has a bandwidth of 0.05 times the sample rate. They will operate at any sample rate. The available center frequencies are 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45. Stopband attenuation is approximately -60 dB and passband ripple is around 1 dB. See Figures 20 and 21.



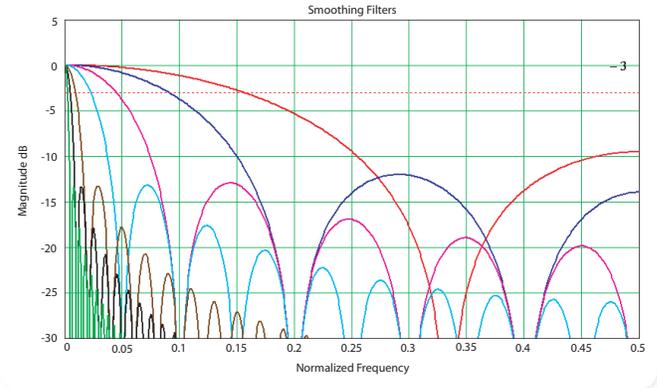
▶ **Figure 22.** Frequency response of the available bandstop filters.

File Name	Normalized Bandwidth	Normalized Center Frequency
bandstop_0.1bw_0.10center.ftt	0.1	0.10
bandstop_0.1bw_0.15center.ftt	0.1	0.15
bandstop_0.1bw_0.20center.ftt	0.1	0.20
bandstop_0.1bw_0.25center.ftt	0.1	0.25
bandstop_0.1bw_0.30center.ftt	0.1	0.30
bandstop_0.1bw_0.35center.ftt	0.1	0.35
bandstop_0.1bw_0.40center.ftt	0.1	0.40

▶ **Figure 23.** Table of available normalized bandstop filters.

Bandstop Filters:

Each filter has a bandwidth of 0.1 times the sample rate. They will operate at any sample rate. The available center frequencies are 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, and 0.40. Stopband attenuation is approximately -110 dB, however, the noise floor of the oscilloscope will not allow for that depth. With an FFT and long

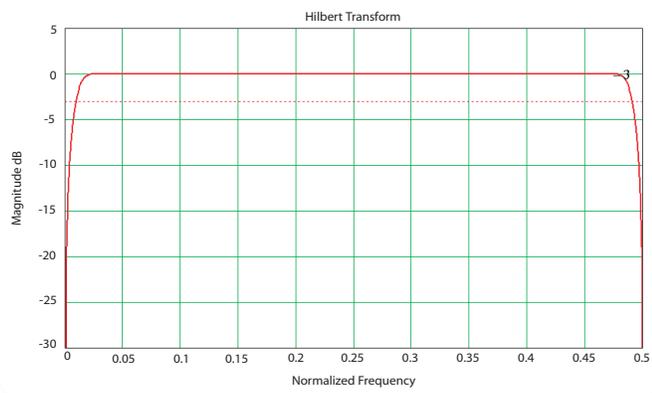


▶ **Figure 24.** Frequency response of the available smoothing filters.

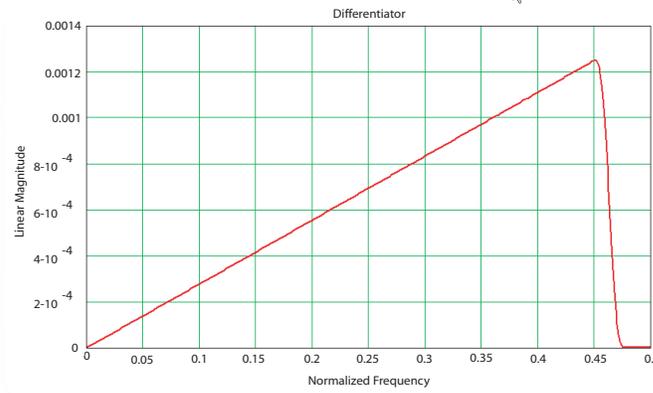
File Name	Length	Normalized Bandwidth	Stopband attenuation dB
smooth3.ftt	3	0.1558	-9.4
smooth5.ftt	5	0.0903	-12
smooth10.ftt	10	0.0446	-12.9
smooth20.ftt	20	0.0224	-13.2
smooth50.ftt	50	0.00887	-13.2
smooth100.ftt	100	0.0045	-13.2
smooth200.ftt	200	0.0022	-13.2

▶ **Figure 25.** Table of available smoothing filters.

record length and averaging turned on one can push noise floors into the -100 dBm range on an 8-bit scope! However, the scope will have some spurious signals above that floor. This is possible because the FFT is an average calculation internally and the averaging function increases the vertical bits of resolution. See Figures 22 and 23.



► **Figure 26.** Frequency response of the Hilbert transform filter.



► **Figure 27.** Frequency response of the differentiator filter.

Smoothing Filters:

These are sometimes called box car filters. They simply average together adjacent samples along the time record. The filter coefficients for these filters are all equal to $1/M$ where M is the length of the filter. The name of the filter files indicate what length the smoothing filter is. Their frequency responses are shown in Figure 24.

Smoothing filters are lowpass filters with a somewhat less than optimal stop band characteristic. However, they are commonly used to remove high frequency noise from a displayed trace. Take care in using to insure that the passband of the signal you are filtering is well within the passband of the filter you choose. That will insure that only noise is removed. Lengths of 3, 5, 10, 20, 50, 100, and 200 are provided in the library. See Figure 25. Refer to Figure 24. The red trace is for filter length 3, followed by blue trace at 5, followed by magenta trace for 10, and so on.

Hilbert Transform Filter:

The ideal Hilbert transform filter has a gain of one at all frequencies and shifts the phase of all frequencies by 90 degrees. This type of filter is one of the types that may be specified in the Remez Exchange algorithm described in “Theory and Application of Digital Signal Processing” referenced on page 15. Notice that this

filter departs from its desired behavior in the frequency range of 0 to 0.025 times the sample rate and also in the range of about 0.475 to 0.5 times the sample rate. This type of filter can be used to create quadrature signals over a wide frequency range. The filename for this filter is HilbertTransform90PhaseShift.flt. See Figure 26.

Differentiator Filter:

The ideal differentiator is a high pass filter that shifts phase by 90 degrees. Its’ frequency response would be linear all the way from DC to 0.5. However, that cannot easily be realized so the response of the filter provided in the library makes a good differentiator for the frequency range of DC to 0.45. See Figure 27.

Section 11: FIR Filter File Format

This section describes the ASCII file format for storing filters for use in the Tektronix oscilloscope waveform math section. A filter menu function allows the user to specify a disk file name containing the filter. A single file format allows the user to specify a different set of coefficients for each sample rate at which the filter is allowed to operate. If the sample rate is not in the file list then the filter will not be applied to the data. The file format also allows the user to specify that the set of filter coefficients is normalized. This allows the same set of filter coefficients to operate at all sample rates.

The ASCII file format is specified as follows:

```
# Comments preceded by # symbol
< sampleRate > coef1, coef2, .... coefN
< sampleRate > coef1, coef2, .... coefN
    |             |
    |             |
< sampleRate > coef1, coef2, .... coefN
```

Each set of filter coefficients in a file are specified in one row preceded by the sample rate value at which that set will operate. If the user specifies the @ symbol for the sample rate then the filter will operate at all sample rates. If the @ symbol is specified then there should only be one set of filter coefficients in the file. However, the user may have other rows with sample rates specified and they will be ignored. There will be a separate row for each sample rate the filter is to operate at. Each row may have a different number of coefficients with a maximum of 1000. The file may contain up to 20 rows.

An example of file content for a normalized filter is shown as follows. This example is the contents of the smooth5.flt file.

```
@    0.2,    0.2,    0.2,    0.2,    0.2
```

An example of a filter that is setup to operate at a specific sample rate is given as follows. This is the contents of a file named 200 MHz_mult_sample_rates.flt that is included in the library directory on the oscilloscope.

```
#This is a 4th order Bessel Thompsen low pass filter.
#200MHz bandwidth, will operate at any of the following sample rates:
# 40 GS/s, 20 GS/s, 10 GS/s, 5 GS/s, 2.5 GS/s, 1 GS/s, 500 MS/s
5e8; 1.968e-007,1.008,-0.00978,0.002267,-0.0002208,1.643e-005,-1.397e-006,1.434e-007
1e9; 9.524e-008,0.3899,0.4877,0.1304,-0.004733,-0.004566,.....
2.5e9; 3.868e-008,0.01885,0.1081,0.1982,0.2284,0.1981,.....
5e9; 1.935e008,0.0007332, 0.009428, 0.02874, 0.05408, 0.07921, .....
1e10; 9.673e-009,3.445e-006,0.0003666,0.001831,0.004714,0.008978,0.01437,0.....
2e10; 4.837e-009, 1.657e-008, 1.723e-006, 4.274e-005, 0.00018334-009, .....
4e10;2.418e-009, -3.524e-009, 8.284e-009, -1.795e-008,8.613e-007, .....
```

Conclusion

This paper has presented information on the design and use of FIR digital filters specifically targeted for the Tektronix oscilloscope waveform math sub-section. A frequency sampling filter design method was described to allow the oscilloscope user to create custom filters. In addition a library of FIR filters that are provided within the oscilloscope was defined. This library provides a wide range of normalized FIR filters for general purpose use. This paper has provided necessary information needed to enable the scope user to make effective use of the arbitrary FIR filter function in a DPO oscilloscope platform math package.

Glossary

Convolution - A process using multiplies and additions to implement a filter operation.

DSP – Digital Signal Processing

DUT – Device Under Test

FIR - Finite Impulse Response

FFT – Fast Fourier Transform

IFFT – Inverse Fast Fourier Transform

IIR - Infinite Impulse Response

Magnitude Response - The magnitude as a function of frequency at the output of a DUT as a function of constant amplitude input as a function of frequency.

Phase Response – the phase as a function of frequency at the output of a DUT given a cosine input signal as a function of frequency with zero phase.

References

1. Alan V. Oppenheim/Rona. W. Schaffer, Digital Signal Processing, Prentice-Hall Inc., Copyright 1975, page 155, ISBN 0-13-214635-5
2. Lawrence R. Rabiner-Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Copyright 1975 page 75-183
3. Lawrence R. Rabiner-Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Copyright 1975 page 187-204

Author: John J Pickerd

Contact Tektronix:

ASEAN / Australasia (65) 6356 3900
Austria +41 52 675 3777
Balkan, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium 07 81 60166
Brazil & South America 55 (11) 3741-8360
Canada 1 (800) 661-5625
Central East Europe, Ukraine and Baltics +41 52 675 3777
Central Europe & Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France & North Africa +33 (0) 1 69 86 81 81
Germany +49 (221) 94 77 400
Hong Kong (852) 2585-6688
India (91) 80-22275577
Italy +39 (02) 25086 1
Japan 81 (3) 6714-3010
Luxembourg +44 (0) 1344 392400
Mexico, Central America & Caribbean 52 (55) 56666-333
Middle East, Asia and North Africa +41 52 675 3777
The Netherlands 090 02 021797
Norway 800 16098
People's Republic of China 86 (10) 6235 1230
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea 82 (2) 528-5299
Russia & CIS 7 095 775 1064
South Africa +27 11 254 8360
Spain (+34) 901 988 054
Sweden 020 08 80371
Switzerland +41 52 675 3777
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

Last Updated January 5 2006

Our most up-to-date product information is available at: www.tektronix.com



Copyright © 2006, Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

1/06 FLG/WWW

4HW-19378-0

Tektronix
Enabling Innovation