

# Reducing FFT Scalping Loss Errors Without Multiplication

“DSP Tips and Tricks” introduces practical design and implementation signal processing algorithms that you may wish to incorporate into your designs. We welcome readers to submit their contributions. Contact Associate Editors Rick Lyons (R.Lyons@ieee.org) or Clay Turner (clay@clayturner.com).

This article discusses the estimation of time-domain sinewave peak amplitudes based on the fast Fourier transform (FFT) data. Such an operation sounds simple, but the scalping loss characteristic of FFTs complicates the procedure. Here we present novel multiplier-free methods to accurately estimate sinewave amplitudes, based on FFT data, that greatly reduce scalping loss problems.

## FFT SCALPING LOSS REVISITED

There are many applications that require the estimation of a time-domain sinewave’s peak amplitude based on FFT data. Such applications include oscillator and analog-to-digital converter performance measurements, as well as standard total harmonic distortion (THD) testing. However, the scalping loss inherent in FFTs creates an uncertainty in such time-domain peak amplitude estimations. This section provides a brief review of FFT scalping loss.

As most of you know, if we perform an  $N$ -point FFT on  $N$  real-valued time-domain samples of a discrete sinewave, whose frequency is an integer multiple of  $f_s/N$  ( $f_s$  is the sample rate in hertz),

the peak magnitude of the sinewave’s positive-frequency spectral component will be

$$M = \frac{A \cdot N}{2}, \quad (1)$$

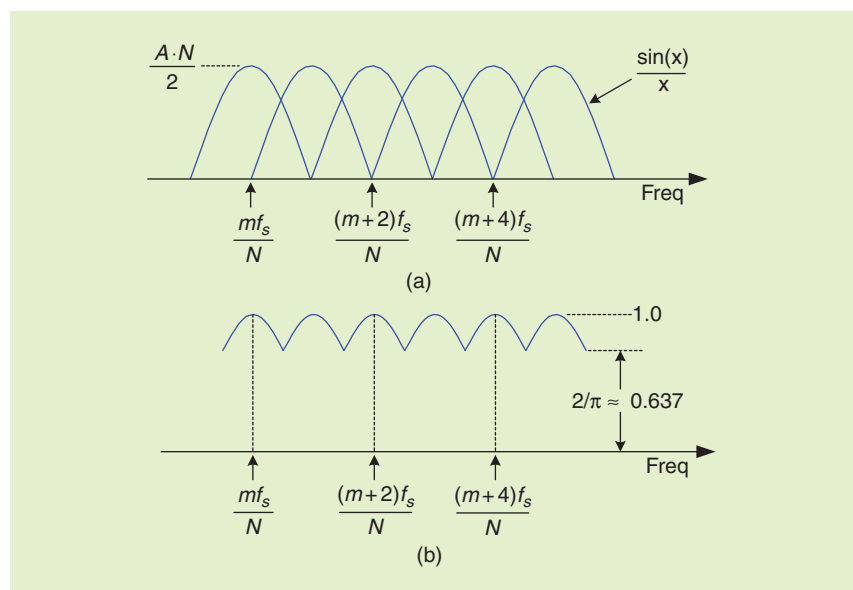
where  $A$  is the peak amplitude of the time-domain sinewave. That phrase “whose frequency is an integer multiple of  $f_s/N$ ” means that the sinewave’s frequency is located exactly at one of the FFT’s bin centers.

Now, if an FFT’s input sinewave’s frequency is between two FFT bin centers (equal to a noninteger multiple of  $f_s/N$ ), the FFT magnitude of that spectral component will be less than the value of  $M$  in (1). Figure 1 illustrates this behavior. Figure 1(a) shows the frequency responses of individual FFT bins where, for simplicity, we show only the main lobes (no side lobes) of the FFT bins’

responses. What this means is that if we were to apply a sinewave to an FFT and scan the frequency of that sinewave over multiple bins, the magnitude of the FFT’s largest normalized magnitude sample value will follow the curve in Figure 1(b). That curve describes what is called the “scalping loss” of an FFT [1].

(As an aside, the word scallop is not related to my favorite shellfish. As it turns out, some window drapery, and tablecloths, do not have linear borders. Rather they have a series of circular segments, or loops, of fabric defining their decorative borders. Those loops of fabric are called scallops.)

What Figure 1(b) tells us is that if we examine the  $N$ -point FFT magnitude sample of an arbitrary-frequency, peak amplitude =  $A$  sinewave, that spectral component’s measured peak magnitude  $M_{\text{peak}}$  can be in anywhere in the range of:



**[FIG1]** FFT frequency magnitude responses: (a) individual FFT bins and (b) overall FFT response.

Digital Object Identifier 10.1109/MSP.2010.939845  
Date of publication: 17 February 2011

$$\frac{0.637 \cdot A \cdot N}{2} \leq M_{\text{peak}} \leq \frac{A \cdot N}{2} \quad (2)$$

depending on the frequency of that sinewave. This is shown as the *rectangular window* curve in Figure 2, where the maximum scalloping error occurs at a frequency midpoint between two FFT bins. The variable  $M$  in Figure 2 is the  $M$  from (1). So if we want to determine a sinewave's time-domain peak amplitude  $A$ , by measuring its maximum FFT spectral peak magnitude  $M_{\text{peak}}$ , our estimated value of  $A$ , from (1), using

$$A = \frac{2M_{\text{peak}}}{N} \quad (3)$$

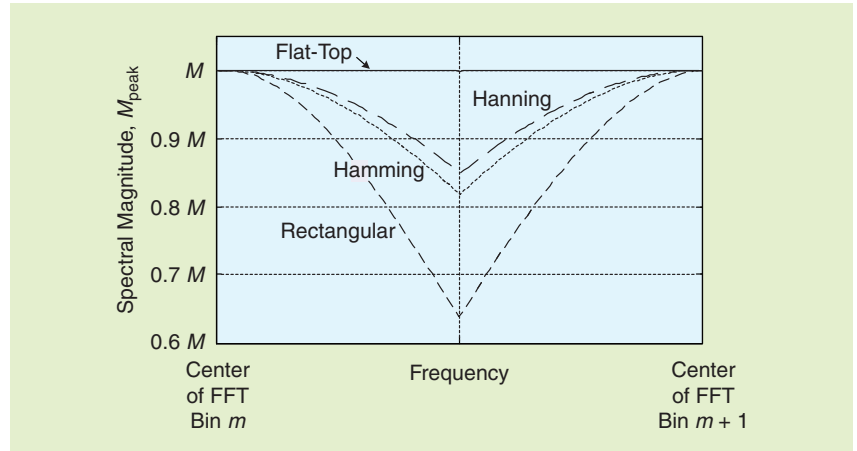
can have an error as great as 36.3%. In many spectrum analysis applications such a large potential error, equivalent to 3.9 dB, is unacceptable. As shown by the magnitude-normalized curves in Figure 2, Hanning and Hamming windowing of the FFT input data reduce the unpleasant frequency-dependent fluctuations in a measured spectral  $M_{\text{peak}}$  value but not nearly enough to satisfy many applications.

One solution to this frequency-dependent, FFT-based, measured amplitude uncertainty is to multiply the original  $N$  time-domain samples by an  $N$ -sample flat-top window function and then perform a new FFT on the windowed data. Flat-top window functions are designed to overcome the scallop loss inherent in rectangular-windowed FFTs. While such a flat-top-windowed FFT technique will work, there are more computationally efficient methods to solve our signal peak amplitude estimation uncertainty problem.

## FREQUENCY-DOMAIN CONVOLUTION

Because multiplication in the time domain is equivalent to convolution in the frequency domain, we can convert rectangular-windowed (no windowing) FFT samples to windowed-FFT samples by way of convolution. For example, consider an  $N$ -point  $w(n)$  window sequences whose time-domain samples are generated using

$$w(n) = \sum_{k=0}^{K-1} (-1)^k h_k \cos(2\pi kn/N), \quad (4)$$



**[FIG2]** Windowed-FFT, bin-to-bin, frequency magnitude responses.

where the  $w(n)$  sequence's generating polynomial has an integer  $K$  number of  $h_k$  coefficients.

Many window functions, including Hanning, Hamming, Blackman, and flat-top, are generated using (4). One popular flat-top window sequence, generated using (4), is Matlab's `flattopwin(N)` routine where the  $h_k$  polynomial coefficients are [2]

$$\begin{aligned} h_0 &= 0.2156, h_1 = 0.4160, \\ h_2 &= 0.2781, h_3 = 0.0836, \\ h_4 &= 0.0069. \end{aligned} \quad (5)$$

(Very similar flat-top window generating coefficients are recommended in [3].) Thus in implementing frequency-domain convolution, to compute a single flat-top windowed  $X_{\text{ft}}(m)$  spectral sample from rectangular-windowed  $X(m)$  spectral samples, we would compute

$$\begin{aligned} X_{\text{ft}}(m) &= \frac{h_4}{2}X(m-4) - \frac{h_3}{2}X(m-3) \\ &+ \frac{h_2}{2}X(m-2) - \frac{h_1}{2}X(m-1) \\ &+ h_0X(m) - \frac{h_1}{2}X(m+1) \\ &+ \frac{h_2}{2}X(m+2) - \frac{h_3}{2}X(m+3) \\ &+ \frac{h_4}{2}X(m+4), \end{aligned} \quad (6)$$

where  $X(m)$  is the rectangular-windowed FFT sample having the largest magnitude, and  $m$  is the FFT's frequency-domain sample index.

If we apply (6) to rectangular-windowed  $X(m)$  FFT samples and compute the flat-top windowed maximum FFT spectral peak

magnitude  $M_{\text{peak}} = |X_{\text{ft}}(m)|$ , the estimated value of  $A$  from (3) will have an error of no more than 0.0166 dB. Such a small error is represented by the very flat, nearly ideal, solid curve labeled as flat-top in Figure 2.

That appealing flat-top curve in Figure 2 is the good news associated with the frequency-domain flat-top window convolution in (6). The bad news is that each computation of an  $X_{\text{ft}}(m)$  sample requires, assuming we combine terms having identical coefficients, 18 real multiplies and 16 real additions. In what follows, we show how to drastically reduce the computational workload in computing an  $X_{\text{ft}}(m)$  sample.

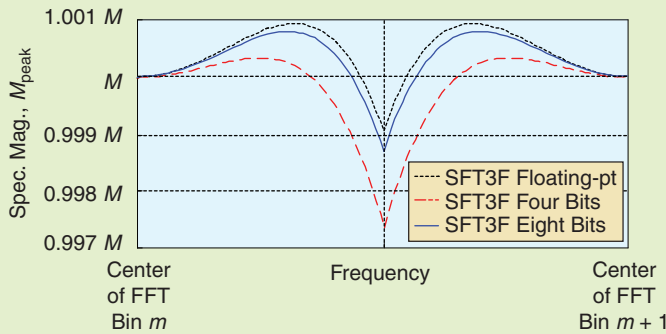
## IMPROVED CONVOLUTION COEFFICIENTS

Reference [4], which discusses many different sets of window generating-polynomial coefficients, presents the following useful set of flat-top window coefficients

$$h_0 = 0.26526, h_1 = 0.5, h_2 = 0.23474 \quad (7)$$

collectively called the SFT3F coefficients. Thus to obtain a single flat-top windowed  $X_{\text{ft}}(m)$  spectral sample from rectangular windowed  $X(m)$  samples, based on the SFT3F coefficients in (7), we compute

$$\begin{aligned} X_{\text{ft}}(m) &= \frac{h_2}{2}X(m-2) - \frac{h_1}{2}X(m-1) \\ &+ h_0X(m) - \frac{h_1}{2}X(m+1) \\ &+ \frac{h_2}{2}X(m+2). \end{aligned} \quad (8)$$



**[FIG3]** Bin-to-bin frequency magnitude response of SFT3F coefficients.

The frequency-dependent scalloping loss of the floating-point SFT3F coefficients is shown as the black dotted curve in Figure 3. The variable  $M$  in Figure 3 is the  $M$  from (1). The computation of an  $X_{fit}(m)$  sample using (8) results in an estimated value of  $A$ , from (3), having a scalloping error in the range of  $-0.0082$  dB to  $+0.0082$  dB. We call the coefficients in (7) “improved” because the computation in (8) requires only ten real multiplies and eight real additions.

Notice that flat-top window coefficients, such as those (7), have the interesting characteristic that they have both a scalloping loss and a scalloping gain versus frequency.

(Compare the black dotted curve in Figure 3 to the lossy Hanning, Hamming, and rectangular curves in Figure 2 whose  $M_{peak}$  values are always less than  $M$ .)

**FURTHER COMPUTATIONAL IMPROVEMENTS**

We can take three steps to further reduce the computational workload of computing an  $X_{fit}(m)$  sample using (8).

**FIRST STEP**

If we divide the coefficients in (7) by the first coefficient,  $h_0$ , we obtain the new coefficients

$$h_0 = 1.0, h_1 = 1.88494, h_2 = 0.88494. \quad (9)$$

The coefficients in (9) eliminate the amplitude gain loss of the flat-top coefficients in (8) without changing their scalloping loss compensation performance. Given the flat-top window generating polynomial coefficients in (9), computing an  $X_{fit}(m)$  sample proceeds as

$$X_{fit}(m) = X(m) - \frac{1.88494}{2} \times [X(m-1) + X(m+1)] + \frac{0.88494}{2} \times [X(m-2) + X(m+2)]. \quad (10)$$

The coefficients in the convolution expression in (10) are

$$g_0 = 1.0, g_1 = -\frac{1.88494}{2} = -0.94247, g_2 = \frac{0.88494}{2} = 0.44247. \quad (11)$$

**SECOND STEP**

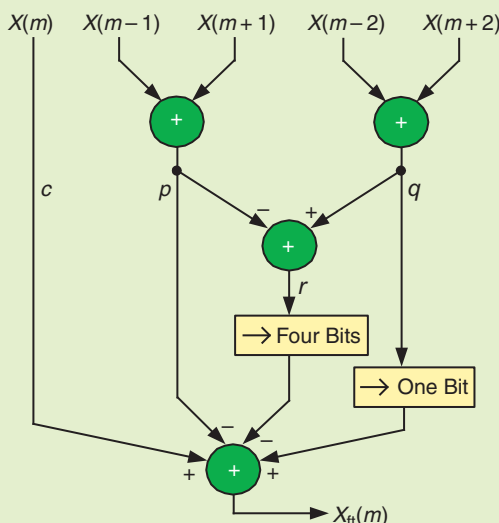
Next, we convert the coefficients in (11) to binary representation to simplify our processing by replacing the multiplications in (10) by arithmetic right-shifts. Doing so, the nonunity coefficients in (11) become

$$g_1 = -0.94247 = -0.1111\ 0001\ 0100\ 0101\dots g_2 = 0.44247 = 0.0111\ 0001\ 0100\ 0101\dots \quad (12)$$

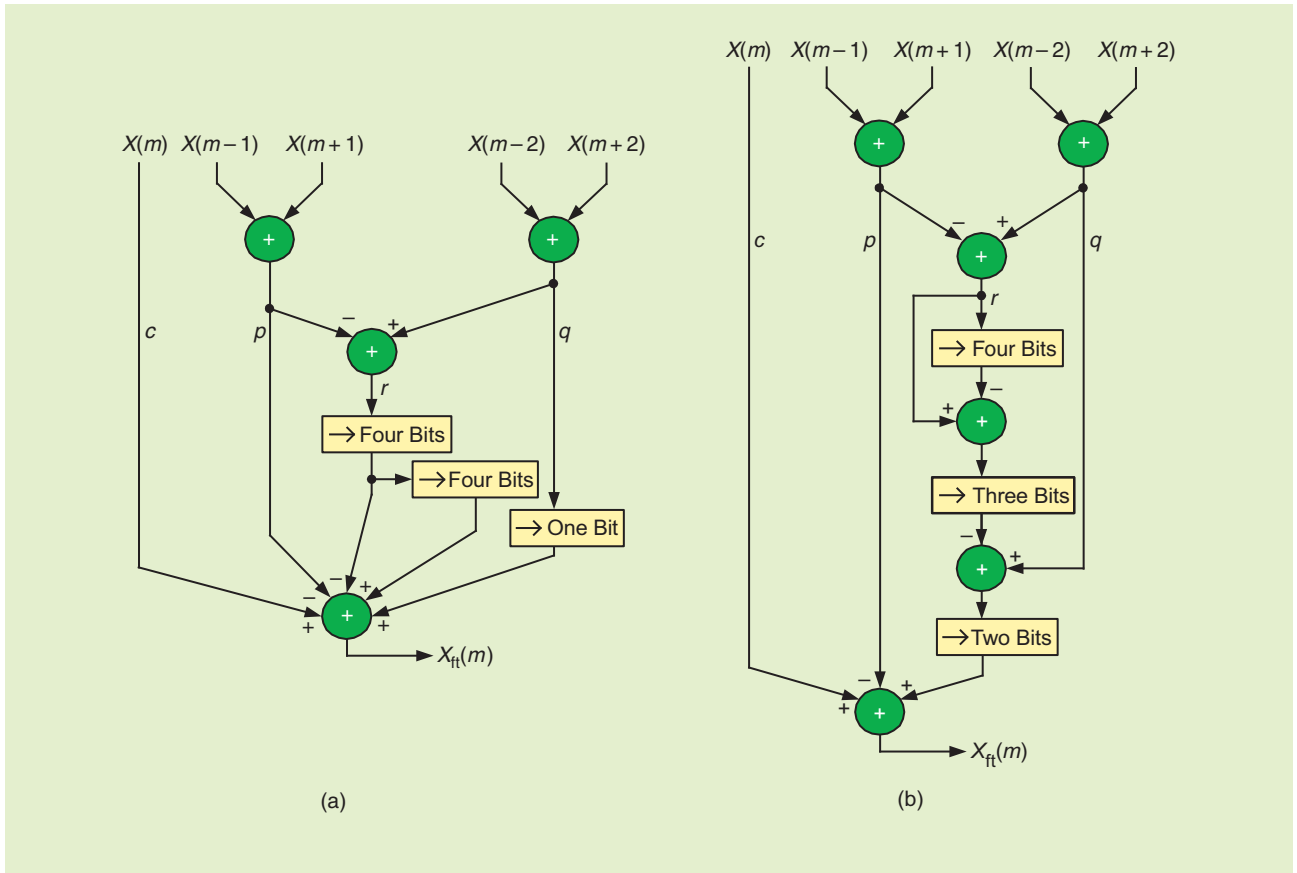
The leftmost sequence of three consecutive zeros in coefficients  $g_1$  and  $g_2$  suggest that we can represent those coefficients with four fractional bits without inducing too much truncation error.

To simplify our equations, let’s represent our five unwindowed frequency-domain samples in (10) with

$$c = X(m) p = X(m-1) + X(m+1) q = X(m-2) + X(m+1) r = q - p.$$



**[FIG4]** Multiplier-free scalloping loss compensation using 4-b coefficients in (14).



**FIG5** Scallop loss compensation using 8-b coefficients: (a) initial implementation and (b) reduced truncation error implementation.

Those assignments convert (10), using the first four fractional bits for  $g_1$  and  $g_2$  in (12), to

$$X_{ft}(m) = c - \frac{1}{2}p + \frac{1}{4}r + \frac{1}{8}r + \frac{1}{16}r, \quad (13)$$

allowing us to replace the multiplications in (10) with binary right-shifts. However, rather than implement the four separate binary right-shifts in (13), we can use canonical signed digit (CSD) notation to further streamline our computations. Using CSD, (13) becomes

$$X_{ft}(m) = c - p + \frac{1}{2}q - \frac{1}{16}r, \quad (14)$$

which is equivalent to, but simpler to compute than, (13). The signal flow implementation of (14) is given in Figure 4, and its performance is shown as the red dashed curve in Figure 3.

Finally, we compute  $M_{\text{peak}}$ , using (14), as

$$M_{\text{peak}} = |X_{ft}(m)| \quad (15)$$

and use that  $M_{\text{peak}}$  value in (3) to compute our desired  $A$ , the peak amplitude of the FFT's time-domain sinewave input. The computation of an  $M_{\text{peak}}$  value using (14) and (15) results in an estimated value of  $A$ , from (3), having a scalloping error in the range of  $-0.0229$  dB to  $+0.003$  dB.

We can achieve 8-b accuracy of our binary coefficients in (12) by adding one more term to the approximation in (14) as

$$X_{ft}(m) = c - p + \frac{1}{2}q - \frac{1}{16}r + \frac{1}{256}r. \quad (16)$$

The signal flow implementation of (16) is given in Figure 5(a), and its performance is shown as the solid blue curve in Figure 3. The computation of an  $X_{ft}(m)$  sample using (16) results in an estimated value of  $A$ , from (15) and (3), having a scalloping error in the range of  $-0.0113$  dB to  $+0.0069$  dB. That's almost worth writing home about because the perfor-

mance of the multiplier-free (16) is superior to the multiply-intensive computation in (6).

### THIRD STEP

In our relentless pursuit of accuracy, we employ one last binary arithmetic trick to reduce right-shift truncation error. Notice in Figure 5(a) that one of our complex data samples experiences a right-shift by 8 b. To reduce the truncation error of an 8-b right shift, we use Horner's rule to convert (16) to

$$X_{ft}(m) = c - p + \frac{1}{2} \left( q - \frac{1}{8} \left( r - \frac{1}{16} r \right) \right). \quad (17)$$

This way, no data sample experiences a truncation error greater than a 4-b right-shift. The signal flow implementation of (17) is given in Figure 5(b) and its performance is equal to that of (16).

To consolidate what we've covered so far, Table 1 shows the computational

[TABLE 1] COMPUTATIONAL WORKLOAD PER  $X_{n(m)}$  SAMPLE AND PERFORMANCE.

SINGLE COMPUTATION EQUATION	REAL MULTS	REAL ADDS	BINARY RIGHT-SHIFTS	MAX. SCALLOPING ERROR (DB)
(6)	18	16	—	0.0166
(8)	10	8	—	0.0082
(10)	4	6	—	0.0082
(14)	—	12	4	0.0228
(16) AND (17)	—	14	6	0.0113

workload, and error performance in estimating a sinewave amplitude  $A$ , of the various scalloping loss compensation methods.

#### IMPLEMENTATION CONSIDERATIONS

There are two issues to keep in mind when using the above scalloping loss compensation methods.

- The flat-top window frequency-domain convolutions are most useful in accurately measuring the time-domain amplitude of a sinusoidal signal when that signal's spectral component is not contaminated by side lobe leakage from a nearby spectral component. For example, if a positive-frequency spectral component is low in frequency, i.e., located in the first few FFT bins, leakage from the spectral component's corresponding negative-frequency spectral component will contaminate that positive-frequency spectral component. As such, empirical testing indicates that the convolutions in Figures 4 and 5 should not be used for frequencies below the sixth FFT bin or above the  $(N/2-5)$ th FFT bin.

- The flat-top window frequency-domain convolutions discussed above are most useful when the FFT spectral component being measured is well above the background spectral noise floor.

#### CONCLUSION

We discussed the inherent scalloping loss uncertainty (potential error) of estimating sinewave peak amplitudes based on FFT spectral data. Then we briefly discussed the performance, and computational workload, of frequency-domain convolution using traditional five-term flat-top window coefficients to drastically reduce sinusoidal peak amplitude measurement uncertainty. Next we demonstrated a little-known three-term flat-top window polynomial that has very good scalloping loss compensation and a reduced computational workload. Finally, we presented a series of binary arithmetic tricks yielding a high-performance, efficient, multiplier-free implementation of scalloping loss compensation. Matlab and C-code implementations of this material are available at: <http://www.signalprocessingsociety.org/>

publications/periodicals/spm/columns-resources/#tips.

#### ACKNOWLEDGMENT

Special thanks go to Clay Turner, holder of a fifth-degree black belt in the engineering of algebra, for his constructive suggestions that improved the content of this article.

#### AUTHOR

*Richard Lyons* (R.Lyons@ieee.org) is a consulting systems engineer and lecturer with Besser Associates in Mountain View, California. He is the author of *Understanding Digital Signal Processing 3/E* (Prentice-Hall, 2010), and editor of, and contributor to, the book *Streamlining Digital Signal Processing, A Tricks of the Trade Guidebook* (IEEE Press/Wiley, 2007).

#### REFERENCES

- [1] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.
- [2] Signal processing toolbox, flat top weighted window. The Mathworks Inc. [Online]. Available: <http://www.mathworks.com/access/helpdesk/help/toolbox/signal/flattopwin.html>
- [3] S. Gade and H. Herlufsen. (1987). Use of weighting functions in DFT/FFT analysis (Part I). *Brüel Kjaer Tech. Rev.* [Online]. (3), pp. 19–21. Available: <http://www.bksv.com/doc/bv0031.pdf>
- [4] G. Heinzel, A. Rüdiger, and R. Schilling. (2002, Feb. 15). Spectrum and spectral density estimation by the discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows. Max-Planck-Inst. für Gravitationsphysik Rep. [Online]. Available: [http://www.rssd.esa.int/SP/LISAPATHFINDER/docs/Data\\_Analysis/GH\\_FFT.pdf](http://www.rssd.esa.int/SP/LISAPATHFINDER/docs/Data_Analysis/GH_FFT.pdf)



## moving?

You don't want to miss any issue of this magazine!

### change your address

**BY E-MAIL:** [address-change@ieee.org](mailto:address-change@ieee.org)

**BY PHONE:** +1 800 678 IEEE (4333) in the U.S.A.  
or +1 732 981 0060 outside the U.S.A.

**ONLINE:** [www.ieee.org](http://www.ieee.org), click on quick links, change contact info

**BY FAX:** +1 732 562 5445

Be sure to have your member number available.