“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 C. Britton Rorabaugh (dspboss@aol.com).

Regarding window functions used in spectral analysis, the most important performance measures are 3-dB bandwidth and sidelobe attenuation. For many window functions, Hanning and Hamming for example, we have no control over a window’s 3-dB bandwidth and sidelobe attenuation for a given window length. For other window functions—Kaiser, Gaussian, and Chebyshev—we can reduce those windows’ 3-dB bandwidth to get improved spectral resolution. However, with these later window functions (what we refer to as “conventional windows”), spectral resolution improvement comes at the expense of sidelobe attenuation reduction that degrades our ability to avoid undesirable spectral leakage. Likewise we can increase those windows’ sidelobe attenuation, but only by sacrificing desirable spectral resolution. This article describes a novel window function that enables us to control both its 3-dB bandwidth (spectral resolution) and sidelobe attenuation (spectral leakage) independently.

The 3-dB bandwidth, sidelobe attenuation, and roll-off rate are used to measure the performance of windows for power spectral density (PSD) estimation [1]–[3]. Improved frequency resolution of the estimated PSD can be obtained if we reduce a window’s 3-dB bandwidth. The sidelobe attenuation means the difference between magnitude of the mainlobe and the maximum magnitude of the sidelobes. The sidelobe roll-off rate is the asymptotic decay rate of sidelobe peaks. Undesirable spectral leakage [4]–[6] can be reduced by increasing sidelobe attenuation and roll-off rate. Therefore, an ideal window for PSD estimation has zero bandwidth and infinite sidelobe attenuation such as an impulse function in frequency domain.

The conventional windows are able to control 3-dB bandwidth or sidelobe attenuation by only one parameter in general [1], [7]–[12]. Thus, they cannot control these two characteristics independently. In other words, if we reduce a window function’s 3-dB bandwidth, the sidelobe attenuation is also reduced, and vice versa [5], [6]. This behavior is the cause of the tradeoff problem between good frequency resolution and acceptable spectral leakage in the estimated PSD. The Butterworth window does not have this problem because it allows control of the 3-dB bandwidth and sidelobe attenuation independently.

Butterworth windows are used as antialiasing filters to reduce the noise in the reconstructed image in previous research [13]. They are also used to remove the edge effect of the matched filter output in pattern matching algorithm [14]. The transfer function of a Butterworth filter is adopted as a window in those applications. However, in this article, a portion of the impulse response of a Butterworth filter is called the Butterworth window and its characteristics in PSD estimation are analyzed.

**BUTTERWORTH WINDOW**
The Butterworth window can be obtained by the standard Butterworth filter design procedure. Important to us is the frequency magnitude response $|H(f)|$ of a Butterworth filter, denoted by [2] and [6]

$$|H(f)| = 1 \sqrt{1 + \left(\frac{f}{f_c}\right)^{2N}},$$

where $f$ is frequency in hertz.

The Butterworth filter is characterized by two independent parameters, 3-dB cutoff frequency $f_c$ and filter order $N$. The cutoff frequency and order of the Butterworth filter serve as parameters that control the bandwidth and sidelobe attenuation of the Butterworth window. The cutoff frequency of a filter has the identical meaning with the bandwidth of a window. However, the cutoff frequency is represented as a half of the bandwidth since the bandwidth of a window refers to two-sided frequency from negative to positive, while the cutoff frequency of a filter refers to only one-sided positive frequency. Our desired window spectrum is identical to the frequency response of the Butterworth filter. Thus, the inverse Fourier transform is applied to the Butterworth filter’s frequency response, in (1), to obtain the filter’s impulse response, and a portion of that response becomes the Butterworth window in the time domain.

**SIMULATION AND PERFORMANCE ANALYSIS**

In our simulation, the frequency and impulse responses of Butterworth filters are investigated to design the Butterworth window by varying the cutoff frequency $f_c$ and filter order $N$. The sampling frequency $f_s$ is set to 2,048 Hz. The magnitude levels of the impulse response of a
Butterworth filter are nearly zero after a certain point—almost all information that determines the filter’s frequency response is in the portion before that zero-magnitude point of the impulse response. Therefore, it is expected that the suitable length of a Butterworth window can be determined by only a part of the infinite-duration impulse response of a Butterworth filter. Figure 1(a) shows the time-domain impulse response $h(k)$ of a unity-gain low-pass Butterworth filter when $f_c = 0.75$ Hz and $N = 3$. In that figure, we show the initial positive-only portion of the impulse response that becomes our desired Butterworth window. The 2,139th sample of $h(k)$ is the point that the magnitude of the impulse response of the Butterworth filter becomes zero for the first time.

The solid curve in Figure 1(b) is the frequency spectrum of the 2,139-sample Butterworth window. The 3-dB bandwidth and sidelobe attenuation of this window are 1.3 Hz and 24.3 dB, respectively. In Figure 1(b), for comparison, we show the frequency magnitude response of the Butterworth filter as the dashed curve. We see that there is no significant difference between the magnitude response of the Butterworth filter and the spectrum of the Butterworth window. Therefore, a suitable length of the Butterworth window may be considered to be up to the point where the magnitude of the impulse response of filter becomes zero for the first time.

Based on the order $N$, the sampling frequency $f_s$, and the cutoff frequency $f_c$ of the Butterworth filter, we have empirically determined the suitable lengths of the Butterworth windows to be those given in Table 1. Here the $[x]$ notation means the integer part of $x$.

The frequency characteristics of Butterworth windows with $f_c = 0.75$ Hz are shown in Table 2. The sidelobe attenuation is increased from ten to 30.4 dB as the filter order is increased from one to five, while the 3-dB bandwidth is fixed at about 1.5 Hz.

The PSD of an example signal is estimated by Butterworth windows to confirm the performance. The signal $x(t)$ used for our simulation is

$$x(t) = 0.84\cos(2\pi \cdot 52 \cdot t) + 0.8\cos(2\pi \cdot 65.5 \cdot t) + 0.3\cos(2\pi \cdot 85 \cdot t) + 1.1\cos(2\pi \cdot 105 \cdot t) + 0.35\cos(2\pi \cdot 140 \cdot t) + 0.98\cos(2\pi \cdot 159 \cdot t) + 0.6\cos(2\pi \cdot 174 \cdot t) + 0.8\cos(2\pi \cdot 190 \cdot t) + \cos(2\pi \cdot 205 \cdot t). \quad (2)$$

The solid lines in Figure 2(a) show the ideal PSD of $x(t)$. The dotted curve in Figure 2(a) shows the estimated PSD of a 2,139-sample rectangular windowed $x(t)$, using Welch’s method [15], where that window’s insufficient sidelobe attenuation (spectral leakage) produces
false spectral components, particularly on either side of a relatively high-level ideal PSD spectral component.

Figure 2(b) shows the estimated PSDs of $x(t)$ using various 2,139-sample Butterworth windows, where we see that reducing the cutoff frequency and increasing the order can reduce spectral leakage without the undesirable spectral mainlobe broadening (loss of resolution) experienced by the conventional window functions. It means the tradeoff problem between resolution and spectral leakage is solved. This beneficial behavior is illustrated in Table 3, where Butterworth windows are compared to the conventional window functions. Reference [16] provides spectral plots comparing Butterworth windows to the conventional window functions.

IMPLEMENTATION ISSUES
- The computational time of PSD estimation is not related to window type, but rather the window length and estimation method. So Butterworth windows have the same computational workload as the conventional window functions.
- To use the computationally efficient radix-2 fast Fourier transform algorithm, we suggest that the time-domain samples of Butterworth window should be zero padded to make the window length an integer power of two. As an alternative to zero padding, we can restrict the Butterworth window’s $f_c$, cutoff frequency to be

$$f_c = \frac{Kf_s}{2^{M+1}}, \quad (3)$$

which leads to Butterworth windows that are $2^M$ in length, where $K$ is one of the scaling constants from Table 1, and $M$ is an integer.
- Because Butterworth windows are not symmetrical, any specialized spectral analysis scheme that requires the imaginary part of a window function’s spectrum to be all zero will not work with the Butterworth windows.

CONCLUSIONS
We’ve shown that the Butterworth window can be obtained by the conventional Butterworth filter design procedure. This window is able to control the 3-dB bandwidth and sidelobe attenuation independently by two parameters, the cutoff frequency and the order of the filter. As such, the sidelobe attenuation can be varied even if the 3-dB bandwidth is fixed, and vice versa. Therefore the tradeoff problem between the frequency resolution and spectral leakage in the estimated PSD, unavoidable with the conventional windows, can be solved by the Butterworth window.

ACKNOWLEDGMENT
The authors thank Associate Editor Richard (Rick) Lyons for his kind help in preparation of this article.

TABLE 3

<table>
<thead>
<tr>
<th>WINDOW</th>
<th>3-DB BANDWIDTH</th>
<th>SIDELOBE ATTENUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECTANGULAR</td>
<td>0.879 Hz</td>
<td>13.3 dB</td>
</tr>
<tr>
<td>TRIANGULAR</td>
<td>1.270 Hz</td>
<td>26.5 dB</td>
</tr>
<tr>
<td>HANNING</td>
<td>1.367 Hz</td>
<td>31.3 dB</td>
</tr>
<tr>
<td>KAISER</td>
<td>$\alpha = 2$</td>
<td>0.980 Hz</td>
</tr>
<tr>
<td></td>
<td>$\alpha = 4$</td>
<td>1.172 Hz</td>
</tr>
<tr>
<td>CHEBYSHEV</td>
<td>$\beta = 1$</td>
<td>0.890 Hz</td>
</tr>
<tr>
<td></td>
<td>$\beta = 2$</td>
<td>1.172 Hz</td>
</tr>
<tr>
<td>BUTTERWORTH</td>
<td>$N = 2$</td>
<td>0.793 Hz</td>
</tr>
<tr>
<td>($f_c = 0.439$ Hz)</td>
<td>$N = 3$</td>
<td>0.740 Hz</td>
</tr>
<tr>
<td></td>
<td>$N = 4$</td>
<td>0.731 Hz</td>
</tr>
<tr>
<td>BUTTERWORTH</td>
<td>$f_c = 0.439$</td>
<td>0.731 Hz</td>
</tr>
<tr>
<td>($N = 4$)</td>
<td>$f_c = 1.500$ Hz</td>
<td>2.815 Hz</td>
</tr>
<tr>
<td></td>
<td>$f_c = 2.500$ Hz</td>
<td>4.720 Hz</td>
</tr>
</tbody>
</table>

AUTHORS
Tae Hyun Yoon (thyoon@ee.knu.ac.kr) is a Ph.D. candidate at the School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Korea.
Eon Kyeong Joo (ekjoo@ee.knu.ac.kr) is a professor at the School of Electrical Engineering and Computer Science,
Kyungpook National University, Daegu, Korea.

REFERENCES