Title:	Computing Data Windows Recursively[1]
Synopsis:	Data windows are indispensible for computing the spectral characteristics of signal waveforms that are not periodic in time. In the context of hardware-based digital signal processing, most if not all data windows are expressed in terms of fairly complex function that are not convenient for embedding in a FPGA or custom ASIC. This memo presents several simple recursive results that can be used to compute these windows without excessive memory requirements and computationally intensive functions.
Author:	James A. Crawford
Date:	30 May 2009

Data windowing prior to performing a FFT is generally necessary in order to reduce the sidelobe response for each frequency bin at the expense of some frequency resolution. Many different windowing function have been developed over the years (e.g., Bartlett, Blackman, Gaussian, Hamming, just to name a few). The section illustrates how the Gaussian and Blackman windowing functions can be recursively generated in time thereby avoiding the need to store large tables for each window function in memory.

1 GAUSSIAN WINDOW

The zero-mean unit-variance Gaussian function is given by

$$g(t) = \frac{\exp\left(-\frac{t^2}{2}\right)}{\sqrt{2\pi}} \tag{1}$$

and its time derivative is

$$g'(t) = -t g(t) \tag{2}$$

Forward Euler integration can be used to formulate the first-order recursion formula for g(t) as

$$\begin{aligned} g[kT_s] &= g\left[(k-1)T_s\right] + g'\left[(k-1)T_s\right]T_s \\ &= g\left[(k-1)T_s\right] + \left\{(-kT_s)\, g\left[(k-1)T_s\right]\right\}T_s \\ &= g\left[(k-1)T_s\right]\left(1 - kT_s^2\right) \end{aligned} \tag{3}$$

If the time increment T_s is sufficiently small, this recursion can be used to generate a Gaussian window with adequate precision as shown in Figure 1.

A more accurate recursion formula for the Gaussian window can be found by making use of the bilinear transform which is equivalent to trapezoidal integration. Although this is still a first-order method, it is an implicit integration method whereas the forward Euler method is not, and this

[1] ©2009, J.A. Crawford. U15199 Calculating Windows Recursively- Notes.pdf, U15198 Calculating Windows Recursively.mcd.

accounts for its substantially better precision. Starting from the same form given by (2), substituting in the z-transform equivalent of d/dt for the derivative produces

$$\frac{2}{T_s}\frac{z-1}{z+1}g = -tg \tag{4}$$

and after collecting terms, this produces the recursion

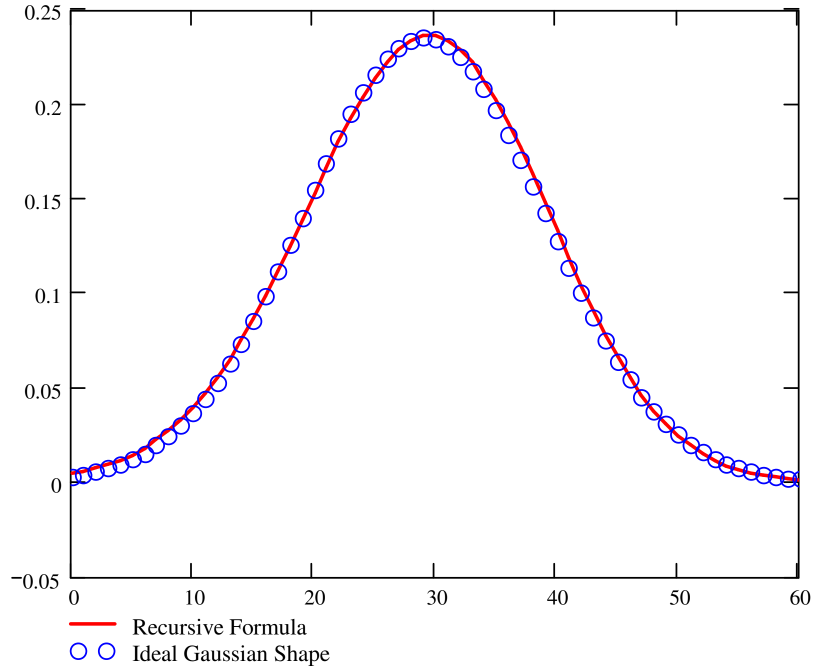

Figure 1 Recursive algorithm based on the forward-Euler integration as given by (3) with T_s = 0.10. Initial starting condition was the true Gaussian value for t_0 = –3.0.

$$g_k = g_{k-1}\frac{\frac{2}{T_s}-(k-1)T_s}{\frac{2}{T_s}+(k-1)T_s} \tag{5}$$

This recursion formula while more accurate, contains a division operation which is not well-suited to digital VLSI integration.

Alternatively, a more accurate recursion formula can be found by using an approximation of the second-order Runge-Kutta method. This second-order form is given by

$$y_{k+1} = y_k + h\, f\left[t_k + \frac{h}{2}, y_k + \frac{h}{2} f\left(t_k, y_k\right)\right] \tag{6}$$

where f represents the analytic derivative of the function being integrated. Following this direction, the resultant recursion is given by

$$g_{k+1} = g_k + \frac{T_s}{2}\left\{-\left(kT_s + \frac{T_s}{2}\right)\left[g_k + \frac{T_s}{2}\left(-kT_s\right)g_k\right]\right\} \tag{7}$$

The apparent accuracy of this recursion formula is equal to or better than that obtained using (5) as shown in Figure 3.

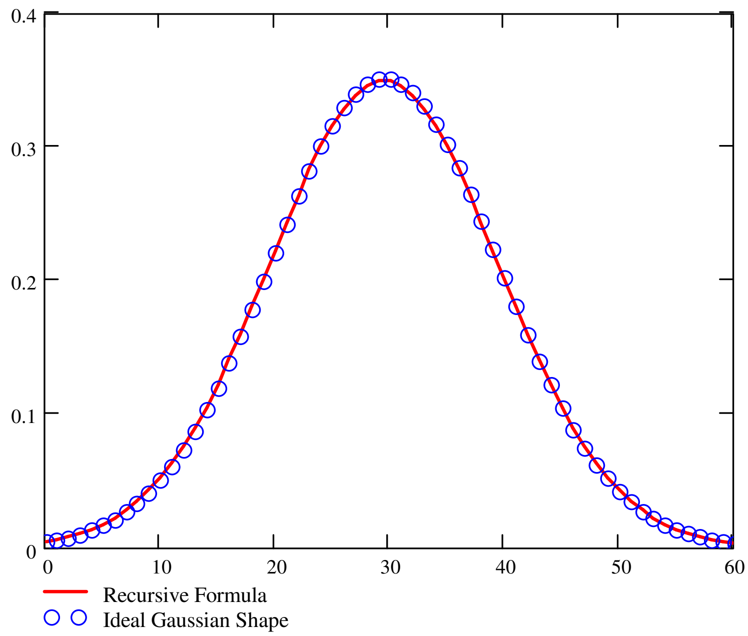

Figure 2 Recursive algorithm given by (5) with T_s = 0.10. Initial starting condition was the true Gaussian value for t_0 = –3.0.

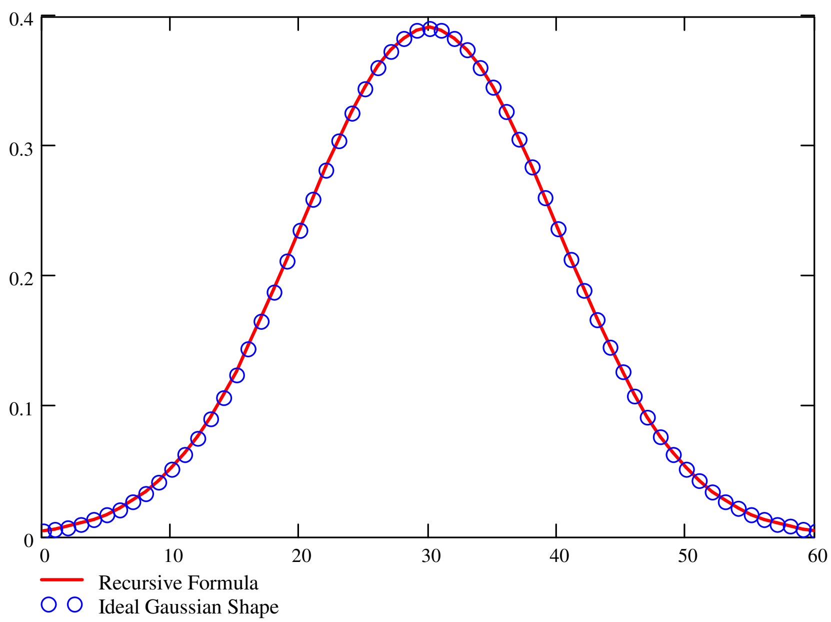

Figure 3 Recursive algorithm given by (7) with T_s = 0.10. Initial starting condition was the true Gaussian value for t_0 = –3.0.

2 BLACKMAN WINDOW

The Blackman data window is given by

$$w(k) = 0.42 + 0.50\cos\left[\pi \frac{2k-1-N}{N-1}\right] + 0.08\cos\left[2\pi \frac{2k-1-N}{N-1}\right] \tag{8}$$

for 1 ≤ k ≤ N. The cosine functions can be created using a recursion based on the second-order differential equation

$$\frac{d^2 y}{dt^2} + \omega_o^2 y = 0 \tag{9}$$

since the solution to (9) is given by $\cos(\omega_o t)$. Use of the bilinear transform for the differential operator can be used to convert (9) into the second-order difference equation given by

$$x_k = c_1 x_{k-1} + c_2 x_{k-2} \tag{10}$$

for which it is assumed that $\omega_o = 1$, $c_2 = -1$, and

$$c_1 = 2\frac{1-\left(\frac{T_s}{2}\right)^2}{1+\left(\frac{T_s}{2}\right)^2} \tag{11}$$

If multiple sine wave cycles need to be computed, better algorithms are available that offer improved stability and absolute periodicity, but this additional precision is not required here. For best results using (10), the seed values should be chosen as $x_0 = \cos(-\Delta\theta)$ and $x_1 = \cos(\Delta\theta)$.

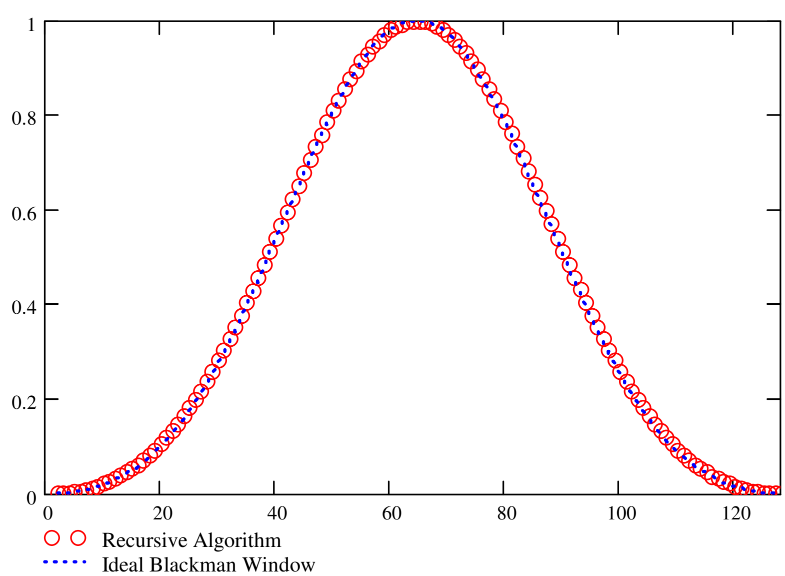

Figure 4 Recursive algorithm versus ideal for the Blackman window function with N = 128

3 SUMMARY

As illustrated by these results for the Gaussian and Blackman windowing functions, recursive methods can be used to create accurate data windows without resorting to extensive table-lookups or painful transcendental functions. The data window values can be computed sample by sample and applied to the serial data that is to be analyzed. This kind of technique can be used for many other window types as well.

