Ronald W. Schafer

What *Is* a Savitzky-Golay Filter?

Recently, while reading a paper on heart rate monitoring using an accelerometer [1], I found myself asking the question posed by the above title. While searching for the answer, I discovered many things that seemed to be well known to others outside the field of digital signal processing (DSP) but not to me. After adding some results of my own, I presented what I had learned at a poster session at the 2011 DSP/SPE Workshop [2]. As I interacted with people at the poster session, I asked if they had ever heard of Savitzky-Golay (S-G) filters. Given my own ignorance, it was comforting that only one out of about 20 had heard of them. What is remarkable about this is that Savitzky and Golay's paper [3], published in 1964, was described in 2000 by editors of the journal *Analytical Chemistry* as number five among the top ten papers ever published in that journal [4]. They stated, "It can be argued that the dawn of computer-controlled analytical chemistry can be traced to this article." For this reason, I feel that it could be useful to use the "Lecture Notes" forum to introduce (or reintroduce) my colleagues in signal processing to the S-G filters.

RELEVANCE

In their "seminal" [4] paper [3], Savitzky and Golay proposed a method of data smoothing based on local least-squares polynomial approximation. They showed that fitting a polynomial to a set of input samples and then evaluating the resulting polynomial at a single point within the approximation interval is equivalent to discrete convolution with a fixed impulse response. The lowpass filters obtained by this method are widely known (in some sectors) as Savitzky-Golay filters. Savitzky and Golay were interested in smoothing noisy data obtained from chemical spectrum analyzers, and they demonstrated that least-squares smoothing reduces noise while maintaining the shape and height of waveform peaks (in their case, Gaussian-shaped spectral peaks). In researching this topic, I did find some awareness of S-G filters in the signal processing community. Hamming's book [7] has a discussion of the use of least-squares in data smoothing, and Orfanidis has a detailed discussion of S-G filters in his book, which is now out of print but available for free download [8]. Some researchers have found the peak shape preservation property of the S-G filters to be attractive in applications such as electrocardiogram processing [1] and the basic concept of least-squares polynomial smoothing has been generalized to two dimensions [5] and applied in processing images such as ultrasound and synthetic aperture radar.

While frequency-domain representations of S-G filters have been discussed [6], [7], most presentations on S-G filters (e.g., [9], [10]) have emphasized time-domain properties (such as complicated closed-form expressions for the impulse responses) without reference to such frequency-domain features as passband width or stopband attenuation. Therefore, the purpose of this article is to examine S-G filters from the frequency-domain viewpoint and to quantify some of their frequency-domain properties.

PREREQUISITES

This article assumes only a familiarity with finite-impulse response (FIR) digital filters and a basic knowledge of matrices.

LEAST-SQUARES SMOOTHING OF SIGNALS

The basic idea behind least-squares polynomial smoothing is depicted in Figure 1, which shows a sequence of samples $x[n]$ of a signal as solid dots. Considering for the moment the group of $2M + 1$ samples centered at $n = 0$, we obtain (by a process to be described) the coefficients of a polynomial

$$p(n) = \sum_{k=0}^{N} a_k n^k \tag{1}$$

that minimize the mean-squared approximation error for the group of input samples centered on $n = 0$,

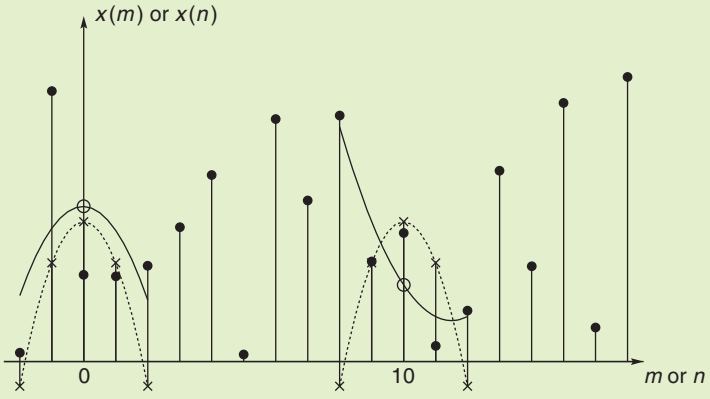
$$\begin{aligned}\mathcal{E}_N &= \sum_{n=-M}^{M} (p(n) - x[n])^2 \\ &= \sum_{n=-M}^{M} \left(\sum_{k=0}^{N} a_k n^k - x[n] \right)^2. \end{aligned} \tag{2}$$

The analysis is the same for any other group of $2M + 1$ input samples. We shall refer to M as the "half width" of the approximation interval. In Figure 1, where $N = 2$ and $M = 2$, the solid curve on the left in Figure 1 is the polynomial $p(n)$ evaluated on a fine grid between -2 and $+2$, and the smoothed output value is obtained by evaluating $p(n)$ at the central point $n = 0$. That is, $y[0]$, the output at $n = 0$, is

$$y[0] = p(0) = a_0, \tag{3}$$

i.e., the output value is just equal to the 0th polynomial coefficient. In general, the approximation interval need not be symmetric about the evaluation point. This leads to nonlinear phase filters, which can be useful for smoothing at

Digital Object Identifier 10.1109/MSP.2011.941097
Date of publication: 15 June 2011



[FIG1] Illustration of least-squares smoothing by locally fitting a second-degree polynomial (solid line) to five input samples: • denotes the input samples, ○ denotes the least-squares output sample, and × denotes the effective impulse response samples (weighting constants). (The dotted line denotes the polynomial approximation to centered unit impulse.)

the ends of finite-length input sequences. The output at the next sample is obtained by shifting the analysis interval to the right by one sample, redefining the origin to be the position of the middle sample of the new block of $2M + 1$ samples, and repeating the polynomial fitting and evaluation at the central location. This can be repeated at each sample of the input, each time producing a new polynomial and a new value of the output sequence $y[n]$. Another example is shown on the right in Figure 1 where the center of the interval is shifted to sample $n = 10$ and the new polynomial fit to the samples $8 \le n \le 12$ is shown again by the solid curve and the output at $n = 10$ is the value of the new polynomial evaluated at the center location.

The original paper by Savitzky and Golay [3] showed that at each position, the smoothed output value obtained by sampling the fitted polynomial is identical to a fixed linear combination of the local set of input samples; i.e., the set of $2M + 1$ input samples within the approximation interval are effectively combined by a fixed set of weighting coefficients that can be computed once for a given polynomial order N and approximation interval of length $2M + 1$. That is, the output samples can be computed by a discrete convolution of the form

$$y[n] = \sum_{m=-M}^{M} h[m]x[n-m] = \sum_{m=n-M}^{n+M} h[n-m]x[m]. \quad (4)$$

The values marked with × in Figure 1 are the shifted impulse responses $h[0-m]$ and $h[10-m]$ that could be used to compute the output samples labeled with ○, thus replacing the polynomial fitting process at each sample with a single evaluation of (4).

To show that we can find a single finite-duration impulse response that is equivalent to least-squares polynomial smoothing for all shifts of the $2M + 1$-sample interval, we must first determine the optimal coefficients of the polynomial in (1) by differentiating $\mathcal{E}_N$ in (2) with respect to each of the $N + 1$ unknown coefficients and setting the corresponding derivative equal to zero. This yields, for $i = 0, 1, \ldots, N$,

$$\frac{\partial \mathcal{E}_N}{\partial a_i} = \sum_{n=-M}^{M} 2n^i \left(\sum_{k=0}^{N} a_k n^k - x[n] \right) = 0, \quad (5)$$

which, by interchanging the order of the summations, becomes the set of $N + 1$ equations in $N + 1$ unknowns

$$\sum_{k=0}^{N} \left(\sum_{n=-M}^{M} n^{i+k} \right) a_k = \sum_{n=-M}^{M} n^i x[n] \quad i = 0, 1, \ldots, N. \quad (6)$$

The equations in (6) are known as the normal equations for the least-squares approximation problem. It is important to note before proceeding that a unique solution requires that we have at least as many data samples as we have coefficients in the polynomial approximation. That is, we require $N \le 2M$. In fact, the equations in (6) become ill-conditioned if M and N are large and N is close to $2M$. Furthermore, if $N = 2M$ the polynomial fits the $2M + 1$ data samples exactly and no smoothing results.

Additional insight can be obtained by expressing the equations in (6) in matrix form. To do this, it is helpful to define a $(2M + 1)$ by $(N + 1)$ matrix $\mathbf{A} = \{\alpha_{n,i}\}$ as the matrix with elements

$$\alpha_{n,i} = n^i, \quad -M \le n \le M, \quad i = 0, 1, \ldots, N.$$

This matrix is called the design matrix for the polynomial approximation problem [10]. The transpose of $\mathbf{A}$ is $\mathbf{A}^T = \{\alpha_{i,n}\}$ and the product matrix $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is an $(N + 1) \times (N + 1)$ symmetric matrix with elements

$$\beta_{i,k} = \sum_{n=-M}^{M} \alpha_{i,n}\alpha_{n,k} = \sum_{n=-M}^{M} n^{i+k} = \beta_{k,i},$$

for $i = 0, 1, \ldots, N$ and $k = 0, 1, \ldots, N$, which we see are the coefficients for the set of equations in (6). Furthermore, if we define the vector of input samples as

$$\mathbf{x} = [x[-M], \ldots, x[-1], x[0], x[1], \ldots, x[M]]^T$$

and define $\mathbf{a} = [a_0, a_1, \ldots, a_N]^T$ as the vector of polynomial coefficients, then it follows that the equations in (6) can be represented in matrix form as

$$\mathbf{Ba} = \mathbf{A}^T\mathbf{Aa} = \mathbf{A}^T\mathbf{x}.$$

Therefore, the solution for the polynomial coefficients can be written as

$$\mathbf{a} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x} = \mathbf{Hx}.$$

Now recall that the output for the group of samples centered on $n = 0$ is $y[0] = a_0$; i.e., we only need to obtain the coefficient a_0. Furthermore, we see

that we only need the 0th row of the $(N+1)\times(2M+1)$ matrix $\mathbf{H}=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$, which by the definition of matrix multiplication gives a_0 as a linear combination of the $(2M+1)$ elements of the $(2M+1)\times 1$ column vector $\mathbf{x}$. The important observation is that the matrix $\mathbf{H}$ depends only on N and M and is independent of the input samples. Thus, the same weighting coefficients will be obtained at each group of $2M+1$ input samples, and so we can think of least-squares smoothing as a shift-invariant discrete convolution process.

One approach to finding the impulse response of the equivalent linear time-invariant (LTI) system is to compute the matrix $\mathbf{H}$. Then, by the definition of matrix multiplication, the output will be

$$y[0]=a_0=\sum_{m=-M}^{M}h_{0,m}x[m],$$

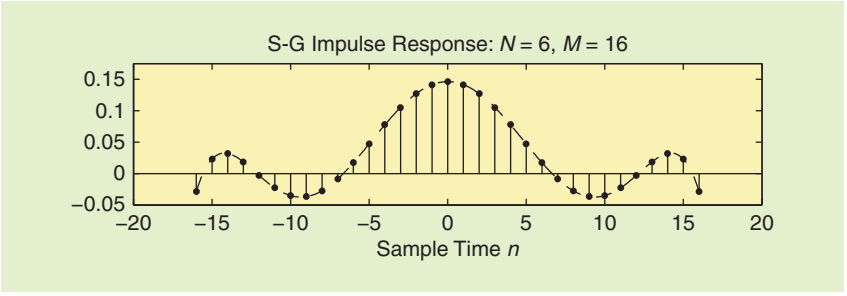
where $h_{i,n}$ denotes the elements of the $(N+1)\times(2M+1)$ matrix $\mathbf{H}$ and $h_{0,m}$ is an element of the 0th row. Therefore, comparing this equation to the second term of (4) with $n=0$, we observe that

$$h[-m]=h_{0,m}\quad -M\le m\le M.$$

Note that this equation gives $h[-m]$ since, as shown in (4), the impulse response is flipped with respect to the input in evaluating discrete convolution. Efficient matrix inversion techniques can be employed to compute only this first row rather than the entire matrix $\mathbf{H}$ [10].

Another approach is to note that since the same weighting coefficients are obtained irrespective of the signal vector, we can set $\mathbf{x}$ equal to a unit impulse centered in the interval $-M\le n\le M$, and solve for all the coefficients of the corresponding polynomial approximation. Note that these polynomial coefficients, denoted as $\widetilde{\mathbf{a}}$, will generally not be equal to those of any of the local approximations that are implicitly generated for each group of $2M+1$ input samples. Then, the impulse response can be obtained by evaluating the corresponding polynomial at locations $-M\le n\le M$.

To show that this statement is true, we denote the coefficient vector for



[FIG2] Impulse response of an S-G filter with $M=16$ and $N=6$. The dashed curve is the polynomial $\tilde{p}(n)$ evaluated on a dense grid.

approximation of the impulse input as $\widetilde{\mathbf{a}}$, which is given by

$$\widetilde{\mathbf{a}}=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{d},$$

where $\mathbf{d}=[0,0,\ldots,0,1,0,\ldots,0,0]^T$ is a $(2M+1)\times 1$ column vector impulse and $\mathbf{A}^T$ is the $(N+1)\times(2M+1)$ matrix

$$\mathbf{A}^T=\begin{bmatrix}(-M)^0&\cdots&(-1)^0&1&1^0&\cdots&M^0\\(-M)^1&\cdots&(-1)^1&0&1^1&\cdots&M^1\\(-M)^2&\cdots&(-1)^2&0&1^2&\cdots&M^2\\\vdots&&\vdots&\vdots&\vdots&&\vdots\\(-M)^N&\cdots&(-1)^N&0&1^N&\cdots&M^N\end{bmatrix}. \quad (7)$$

Then for the impulse input $\mathbf{d}$, it follows that $\mathbf{A}^T\mathbf{d}$ is the $(N+1)\times 1$ column vector

$$\mathbf{A}^T\mathbf{d}=[1,0,\ldots,0]^T.$$

This means that the symmetric matrix $(\mathbf{A}^T\mathbf{A})^{-1}$ must have the form

$$(\mathbf{A}^T\mathbf{A})^{-1}=\begin{bmatrix}\widetilde{a}_0&\widetilde{a}_1&\cdots&\widetilde{a}_N\\\widetilde{a}_1&\bullet&\cdots&\bullet\\\vdots&\vdots&\vdots&\vdots\\\widetilde{a}_N&\bullet&\cdots&\bullet\end{bmatrix},$$

where the matrix entries denoted $\bullet$ do not enter into the computation of $\widetilde{\mathbf{a}}$. Now, since $\mathbf{A}^T$ is as given in (7), it follows from the definition of matrix multiplication that the 0th row of the matrix $\mathbf{H}=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is

$$[h_{0,-M},h_{0,-M+1},\cdots,h_{0,0},\cdots,h_{0,M}]$$
$$=[\tilde{p}(-M),\tilde{p}(-M+1),\cdots,\tilde{p}(0),\cdots,\tilde{p}(M)],$$

where $\tilde{p}(n)$ is the polynomial fit to the unit impulse evaluated at the integers $-M\le n\le M$,

$$\tilde{p}(n)=\sum_{k=0}^{N}\widetilde{a}_k n^k\quad -M\le n\le M. \quad (8)$$

Therefore, the impulse response of the S-G filter is

$$h[-n]=h_{0,n}=\tilde{p}(n).$$

As before, this equation gives $h[-n]$ since the impulse response is flipped around $n=0$ in evaluating discrete convolution. Henceforth, we shall refer to $\tilde{p}(n)$ as the impulse response design polynomial. As we will discuss in a later section, (8) is the basis for a simple method for the design of S-G filters using the polynomial fitting functions in MATLAB.

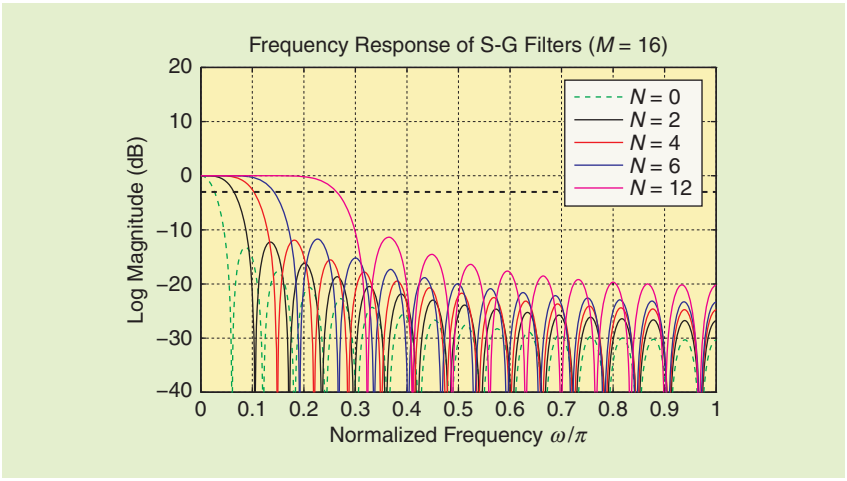
PROPERTIES OF S-G FILTERS

Figure 2 shows the impulse response of an S-G filter with $N=6$ and $M=16$. Although this is a specific example, its time-domain properties are representative of the entire class of symmetric S-G filters.

Figure 3 shows the frequency response of several S-G filters designed by MATLAB statements given in the section "Design of S-G Filters." The impulse response lengths are all $(2M+1)=2\cdot 16+1=33$ with implicit polynomial orders of $N=0,2,4,6,12$. Figures 2 and 3 illustrate properties shared by all S-G filters. These properties, which result from the structures of the matrices $\mathbf{B}$ and $\mathbf{H}$, are summarized below:

- P1 The odd-indexed coefficients of the impulse response design polynomial are all zero so that we can express $\tilde{p}(n)$ as

$$\tilde{p}(n)=\sum_{k=0}^{\lfloor N/2\rfloor}\widetilde{a}_{2k}n^{2k}, \quad (9)$$



[FIG3] **Frequency response of S-G filters for $M = 16$ and various polynomial orders.**

where $\lfloor\ \rfloor$ means rounding down. Among other things, this means that the S-G filters for N and $N+1$ are identical for N an even integer.

■ **P2** Moving average (MA) filtering defined as

$$y[n] = \frac{1}{2M+1} \sum_{m=n-M}^{n+M} x[m]$$

is identical to S-G smoothing with polynomials of order $N = 0$ (constant) and $M = 1$ (straight line).

■ **P3** The impulse response is symmetric since $h[-n] = \tilde{p}(n) = \tilde{p}(-n) = h[n]$. Therefore, the frequency response is purely real. (The shifted impulse response $h[n-M]$ is causal and the corresponding frequency response has linear phase corresponding to the time delay of M samples.) S-G filters are type I FIR lowpass filters [11] with nominal passband gain of unity.

■ **P4** The zeros of the system function $H(z)$ of an S-G filter are either on the unit circle of the z-plane or they occur in complex conjugate reciprocal groups [11]. The unit circle zeros are, of course, responsible for the sharp dips (high attenuation) in the stopband of the frequency responses in Figure 3.

■ **P5** S-G filters have very flat frequency response in their passbands since it can be shown using (8) and (9) and the normal equations (6) that $H(e^{j\omega})|_{\omega=0} = 1$ and

$$\left.\frac{d^r H(e^{j\omega})}{d\omega^r}\right|_{\omega=0} = (-j)^r \sum_{n=-M}^{M} n^r h[n] = 0, \tag{10}$$

for $r = 1, 2, \ldots, N$. Furthermore, it can be shown using the product rule from differential calculus and Parseval's theorem that (10) guarantees that the first N moments of the input signal $x[n]$ are preserved in the output $y[n]$; i.e.,

$$\sum_{n=-\infty}^{\infty} n^r y[n] = \sum_{n=-\infty}^{\infty} n^r x[n] \qquad r = 1, 2, \ldots, N. \tag{11}$$

■ **P6** The nominal normalized cutoff (3 dB down) frequency, $f_c = \omega_c/\pi$, depends on both the implicit polynomial order N and the length of the impulse response, $(2M+1)$. If M is fixed as in Figure 3, the passband of the filter gets wider approximately in proportion to N. Although not illustrated in Figure 3, the cutoff frequency also depends inversely on M. S-G filters are often compared against a MA filter with the same impulse response length [10]. Figure 3 shows that this is somewhat unfair since a shorter MA filter could have roughly the same cutoff frequency as a longer S-G filter with higher value of N. To clarify this interaction of N and M, the next section gives an approximate empirical relation for f_c as a function of both N and M.

■ **P7** The S-G filters have mediocre attenuation characteristics in their stopband regions (except at the frequencies corresponding to zeros on the unit circle). Defining the stopband as the frequency range from the first zero up to π radians, we see from Figure 3 that for the MA filter ($N = 0$ or 1), the minimum attenuation in the stopband (amplitude of first peak after the first zero) is approximately 13 dB. For $N \geq 2$, the minimum attenuation in the stopband is approximately 11 dB. Figure 3 also shows that the peak stopband gain tends to increase with increasing N for fixed M and that the frequency response decreases slightly in gain as frequency increases above the nominal cutoff frequency.

DESIGN OF S-G FILTERS

Recall from (8) that the impulse response of an S-G filter can be computed as samples of the Nth degree polynomial fit to the unit impulse sequence. This method of computing the S-G filters is easily implemented using MATLAB's polynomial functions as in the following MATLAB statements:

```
a=polyfit(-ML:MR,...
    [zeros(1,ML),1,zeros
    (1,MR)],N);
h=fliplr( polyval(a,-ML:MR))
```

The MATLAB function `polyfit()` computes the coefficients of the impulse response design polynomial and `polyval()` evaluates the polynomial at a discrete set of points. Note that these statements can be used to compute nonsymmetric S-G filters by setting `ML≠MR`. The MATLAB Signal Processing Toolbox has a function `sgolayfilt()` for designing and implementing both symmetric and nonsymmetric S-G filters.

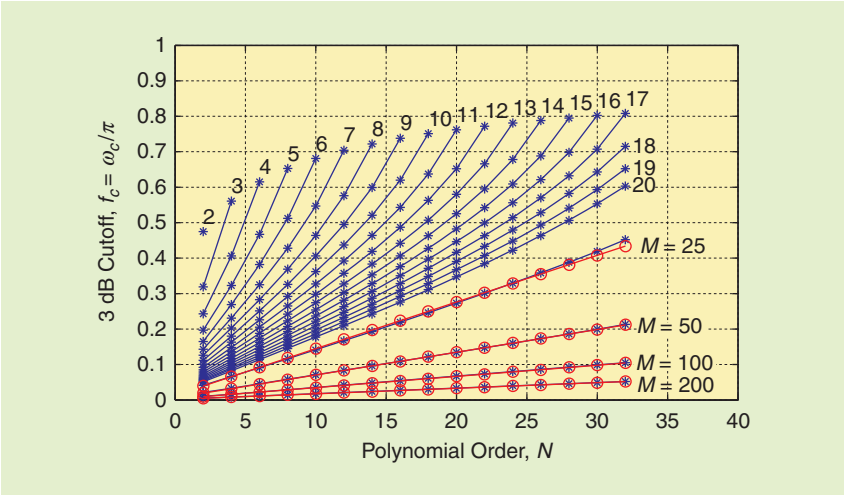
There are some important constraints in the use of polynomial fitting in general. Specifically, the number of data points (in this case $2M+1$) must be strictly greater than the number of undetermined coefficients $N+1$ to achieve smoothing

by the S-G process. Furthermore, if the order of the polynomial, N, is too large, the approximation problem is badly conditioned and the solution will be of no value. (The function `polyfit()` issues an alert when the approximation problem is ill conditioned.) Although these factors are significant limitations, a wide range of frequency-domain characteristics can be achieved nevertheless by choosing M and N appropriately.

To quantify the frequency-domain behavior of S-G filters, impulse responses were computed for various values of M and N within the constraints mentioned above, and the corresponding frequency responses were computed for $0 \le \omega \le \pi$. The passband of the filter was defined by the frequency where $20 \log_{10}|H(e^{j\omega})|$ is "3 dB down" from the value of 0 dB, the gain of the filter at $\omega = 0$. The results for measurements on filters with $M = 2, 3, \ldots, 20$ and $M = 25, 50, 100, 200$ for even polynomial orders N are displayed in Figure 4. The points marked with * and connected by a blue line are the measured cutoff frequencies for a fixed value of M. The values for commonly used short filters ($2 \le M \le 6$ and $N < 2M$) are given more precisely in Table 1. These are the cutoff frequencies for all possible symmetric S-G filters for impulse response lengths $5 \le (2M + 1) \le 13$. Observe that the cutoff frequencies that are achievable range from 0.165 to 0.681, but only a discrete set of values is possible.

In all cases in Figure 4, f_c varies almost linearly with N when $N \ll 2M$ with the slope being dependent inversely on M, but the curves for $M < 25$ tend to deviate from a straight line as N increases toward $2M$. However, when M is large, as in the four cases $M = 25, 50, 100, 200$, the linear region of the curve coincides with the range of usable values of N, so a nearly linear relation holds over a wide range of N. A reasonably accurate approximation to this behavior for the indicated range of parameters is the equation

$$f_c = \frac{N+1}{3.2M - 4.6} \quad M \ge 25 \text{ and } N < M. \tag{12}$$



[FIG4] Relationship between 3 dB cutoff frequency $f_c = \omega_c/\pi$, polynomial length N, and impulse response half-length M.

The values of $f_c = \omega_c/\pi$ predicted by this equation are marked with a red ∘ and connected by a red line. Figure 4 shows that this simple formula fits the measurements quite well even for the case $M = 25$ where the measurements deviate only slightly from the straight line over the entire range of N. The relative error in predicting the measured cutoff frequency is less than 4% over the range $M = 25, 50, 100, 200$ and $N = 4, \ldots, 32$, and the relative error is within 8% for the cases $M = 25,50, 100, 200$ and $N = 2$. As can be seen, large values of M and small N lead to extremely narrow passbands, which would be of limited usefulness except when the signal components are greatly oversampled. Even though the function `polyfit()` gave an ill-conditioned warning for the larger values of N, the resulting filters remained acceptable for values of N up to about 40. The formula in (12) becomes increasingly accurate for larger values of M and N. The formula does not fit as well for values of M less than 25. However, the dependence of f_c on N is still linear except for small M. For $10 \le M < 25$ and N suitably restricted, a formula similar to (12) with 4.6 replaced by 2 gives more accurate predictions. While a more complicated functional form based on more measurements could provide more accurate predictions over a wider range of M and N, (12) should be adequate for most applications of S-G filters where precise specification of the cutoff frequency is not required.

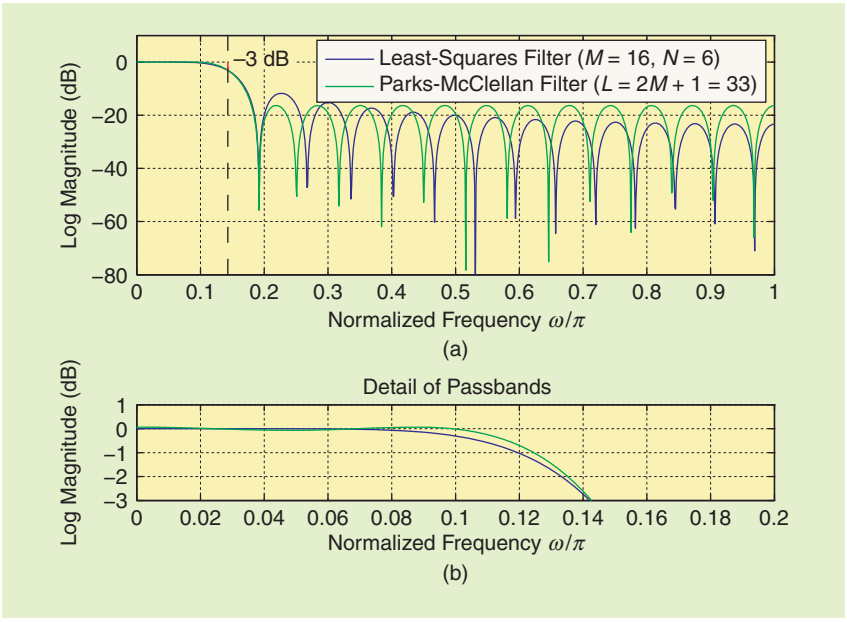
[TABLE 1] NORMALIZED 3 dB CUTOFF FREQUENCIES $f_c = \omega_c/\pi$ AS A FUNCTION OF M AND N.

	POLYNOMIAL ORDER, N				
M	2	4	6	8	10
2	0.475	–	–	–	–
3	0.319	0.561	–	–	–
4	0.243	0.406	0.615	–	–
5	0.197	0.323	0.467	0.653	–
6	0.165	0.269	0.382	0.512	0.681

Figure 4 points out another feature of S-G filters that is often overlooked. A given desired cutoff frequency f_c can be realized by different combinations of N and M. For example, we can achieve a cutoff frequency $f_c = \omega_c/\pi \approx 0.4$ using the (N, M) pairs (4,4), (9,10), (14,16), and (22,19). These filters differ in the sharpness of their transition from flat passband to stopband, which just reflects a familiar property of FIR discrete-time lowpass filters that the widths of transition regions are typically inversely proportional to the length of the impulse response.

DISCUSSION

While there is value in knowing that a single S-G impulse response implicitly achieves local polynomial fitting for every output sample, in many



[FIG5] Comparison of an S-G filter ($N = 6$ and $M = 16$) with an equal-length equiripple filter designed with the PM algorithm. (a) Entire frequency response and (b) the passband region.

applications, signals are not characterized in terms of their ability to be modeled by polynomials but rather in terms of their frequency spectra. Thus, we have focused in this article on the frequency-domain properties of the S-G filters.

S-G filters are often preferred (even revered in some circles) because, when they are appropriately designed to match the waveform of an oversampled signal corrupted by noise, they tend to preserve the width and height of peaks in the signal waveform. While such performance features are often explained in terms of matching fitted polynomial slopes to signal slopes or to the preservation of signal moments, the reason for this behavior is more obvious from the frequency-domain properties of the filters. Specifically, S-G filters have extremely flat passbands with modest attenuation in their stopbands. Furthermore, the symmetric S-G filters have zero phase so that features of the signal are not shifted. Thus, if the signal has most of its energy in the filter passband (implying significant over-sampling), the signal components are undistorted while some high-frequency noise is reduced but not completely eliminated. Of course, assuming that the signal is lowpass and oversampled is equivalent to assuming that the signal is "smooth enough" to be represented by a polynomial of "high enough" degree. However, S-G filters are often used in applications where a direct frequency-domain specification is more precise or more easily related to models for signal production. In such cases, the empirical relationship in (12) or the plot of Figure 4 may be useful. Even in the case of the sampled Gaussian wavelets that model chemical spectrum lines, the corresponding Fourier transform also has Gaussian shape, and it is straightforward to determine the frequency-domain width as a function of the width of the Gaussian wavelet.

From the frequency-domain point of view, the question naturally arises as to whether the main desirable property of the S-G filters (very flat passband) could be achieved with another design method, and perhaps with greater attenuation in the stopband region. Figure 5 shows the frequency response of an S-G filter with $M = 16$ (impulse response length $L = 2M + 1 = 33$) and $N = 6$. Also shown is the frequency response of a length $L = 33$ filter designed by the Parks-McClellan (P-M) algorithm. In this example, the passband and stopband cutoff frequencies of the P-M filter were adjusted by trial and error so that the transition region and the location of the first zero of the frequency response were approximately in the same location as those of the corresponding S-G filter. The measured 3 dB cutoff frequency of the S-G filter was $f_c = 0.143$ (the formula of (12) predicts $f_c = 0.15$). A very flat passband is achieved with the P-M design algorithm by imposing a 10:1 weighting ratio between the passband equiripple approximation error and the stopband approximation error. Larger ratios will make the passband even flatter. In the case of the S-G filter, the gain at the first local maximum beyond the first zero of the frequency response is -11.73 dB, while the equiripple maxima of the P-M filter have gains of -19.9 dB. The lower part of the plot shows that the passband gain of the P-M filter has small ripple about 0 dB, and the flat region is in fact wider than that of the S-G filter. However, due to the tendency of S-G frequency responses to fall off at high frequencies, the S-G filter has slightly lower peak stopband gain than the P-M filter after about $\omega/\pi = 0.5$.

Given the close similarity of the two frequency responses in Figure 5, it is clear that for the case of a signal confined to the band $|\omega| < 0.143\pi$ with additive white noise, the performance of the two systems will be nearly identical. Experiments with longer impulse responses show that P-M filters can achieve very flat passbands and greater stopband attenuation than an equivalent S-G filter. Also, recall that the cutoff frequencies of the S-G filters are restricted to a finite set while those of the P-M filter are not.

WHAT WE HAVE LEARNED

This article has attempted to answer the question "What is a Savitky-Golay filter?" in terms that will be familiar to the DSP community and readers of *IEEE Signal Processing Magazine*. We reviewed the definition and properties of S-G filters and showed how they can be designed easily using polynomial approximation of an impulse sequence. In contrast to most discussions of S-G filters, we focused on the frequency-domain properties, and offered an

approximate formula for the 3-dB cutoff frequency as a function of polynomial order N and impulse response half-length M. Engineers with a frequency-domain mindset (like the author) may find this useful if they choose to use S-G filters in their application.

AUTHOR

Ronald W. Schafer (ron.schafer@hp.com) is an HP Fellow in the Mobile and Immersive Experience Lab at HP Labs, Palo Alto, California, where he is involved in research on acoustic and audio signal processing. From 1974 to 2004, he was John and Marilu McCarty Professor of the School of Electrical and Computer Engineering at Georgia Tech. He is the coauthor of several DSP textbooks including *Discrete-Time Signal Processing* (with Oppenheim), *Signal Processing First* (with McClellan and Yoder), and *Theory and Applications of Digital Speech Processing* (with Rabiner).

REFERENCES

[1] K. Pandia, S. Revindran, R. Cole, G. Kovacs, and L. Giaovangrandi, "Motion artifact cancellation to obtain heart sounds from a single chestworn accelerometer," in *Proc. ICASSP-2010*, 2010, pp. 590–593.

[2] R. W. Schafer, "On the frequency-domain properties of Savitzky-Golay filter," in *Proc. 2011 DSP/SPE Workshop*, Sedona, AZ, Jan. 2011, pp. 54–59.

[3] A. Savitzky and M. J. E. Golay, "Soothing and differentiation of data by simplified least squares procedures," *Anal. Chem.,* vol. 36, pp. 1627–1639, 1964.

[4] J. Riordon, E. Zubritsky, and A. Newman, "Top 10 articles," *Anal. Chem.,* vol. 72, no. 9, pp. 324A–329A, May 2000.

[5] M. Sühling, M. Arigovindan, P. Hunziker, and M. Unser, "Multiresolution moment filters: Theory and applications," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 484–495, Apr. 2004.

[6] M. U. A. Bromba and H. Ziegler, "Application hints for Savitzky-Golay smoothing filters," *Anal. Chem.*, vol. 53, no. 11, pp. 1583–1586, Sept. 1981.

[7] R. W. Hamming, *Digital Filters*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[8] S. J. Orfanidis. (1995–2009). Introduction to signal processing [Online]. Available: www.ece.rutgers.edu/~orfanidis/intro2sp

[9] P.-O. Persson and G. Strang, "Smoothing by Savitzky-Golay and Legendre filters," *IMA Vol. Math. Systems Theory Biol., Comm., Comp., and Finance*, vol. 134, pp. 301–316, 2003.

[10] W. H. Press, S. A. Teukolsky, W. T. Vertterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. Cambridge, U. K.: Cambridge Univ. Press, 2007.

[11] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* 3rd ed. Upper Saddle River, NJ: Pearson, 2010.

[SP]

Kivanc Kose and A. Enis Cetin

Low-Pass Filtering of Irregularly Sampled Signals Using a Set Theoretic Framework

In this article, the goal is to show that it is possible to filter nonuniformly sampled signals according to specs defined in the Fourier domain. In many practical applications, it is necessary to filter irregularly sampled data including seismic signal processing, synthetic aperture radar (SAR) imaging systems, three-dimensional (3-D) meshes, and digital terrain models [1], [2]. In almost all of these practical problems, it is possible to define the desired filtering solution in a set theoretic framework. This lecture note presents a new method for filtering irregularly sampled data by defining stopband tolerance regions in the Fourier domain and time-domain upper and lower bounds on the signal samples as a part of the filtering process. Since there are specifications in both time and frequency domains, it is possible to iterate between time and frequency domains using the fast Fourier transform (FFT) while imposing the constraints in each domain.

Digital Object Identifier 10.1109/MSP.2011.941098
Date of publication: 15 June 2011

RELEVANCE

The ideas presented here can be used to develop filtering algorithms for irregularly sampled one or higher dimensional data. It can be used as a teaching material in advanced undergraduate and graduate discrete-time signal processing, optimization as well as applied mathematics courses.

PREREQUISITES

The prerequisites for understanding this article's material are linear algebra, discrete-time signal processing, and basic optimization theory.

PROBLEM STATEMENT

Let us assume that samples $x_c(t_i)$, $i = 0, 1, 2, \ldots, L-1$, of a continuous time-domain signal $x_c(t)$ are available. These samples may not be on an uniform sampling grid. Let us define $x_d[n] = x_c(nT_s)$ as the uniformly sampled version of this signal. We assume that the sampling period T_s is sufficiently small (below the Nyquist period) for the signal $x_c(t)$. In a typical discrete-time filtering problem, we have $x_d[n]$ or its noisy version, and we apply a discrete-time low-pass filter to the uniformly sampled signal $x_d[n]$. However, $x_d[n]$ is not available in this problem. Only nonuniformly sampled data $x_c(t_i)$, $i = 0, 1, 2, \ldots, L-1$ are available in this problem.

GOAL

Our goal is to low-pass filter the nonuniformly sampled data $x_c(t_i)$ according