ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850
607-266-8492

APPLICATION NOTE NO. 368

December 2006

INTERPOLATION WITH A MINIMUM NORM POLYNOMIAL

INTERPOLATION – LINEAR AND (*intfilt*) PARABOLIC CASES

We are familiar with many forms of signal interpolation, and we know that these are basically a low-pass filtering of a sequence of known samples. Fig. 1 shows a typical, simple case where the signal model is piecewise linear.

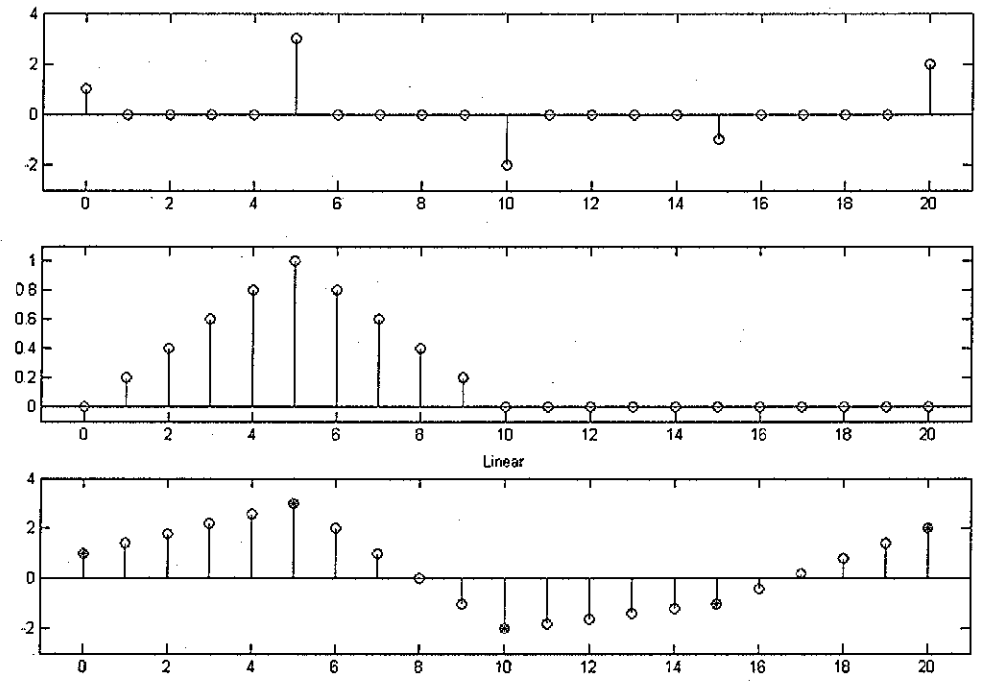

Fig. 1 Piecewise Linear Interpolation by a factor of 5

In Fig. 1 the top shows a given sequence that has been zero-padded by a factor or 5 (that is to say, four zeros have been inserted between each original sample). We want to replace these zeros by interpolated values, and the bottom view shows the case where the interpolated samples are assumed to be on straight lines between the known samples. The interpolation is done by the FIR filter that has the impulse response shown in the middle of Fig. 1. Note that the impulse response value are h_{lin}={1/5 2/5 3/5 4/5 5/5 4/5 3/5 2/5 1/5}. These we can obtain in several ways – the easiest is to just ask the question: what is the impulse response of a linear interpolator. The answer is: the response of a linear interpolator to an impulse. Trick question – but it tells us exactly what to do [1,2].

We note two things about this linear interpolation. First, it involves only two consecutive samples. Secondly, the segments are (be definition) straight lines and thus generally have a discontinuous derivative at the original sample points. This leads us to worry a bit that the method may not be well suited for common signals that we suspect are more bandlimited (more rounded). For a more comfortable result, we might well consider frequency-domain bandlimited interpolation (which would be sinc interpolation in the time domain) instead of our linear segments (first-order polynomial interpolation). Intermediate cases may well be provided by time-domain interpolation with higher-order polynomials. This more general polynomial interpolation (Matlab's ***intfilt***) is very well studied [2], and we will next look at this case of parabolic interpolation.

Fig. 2 shows the case where we use parabolic rather than linear interpolation:

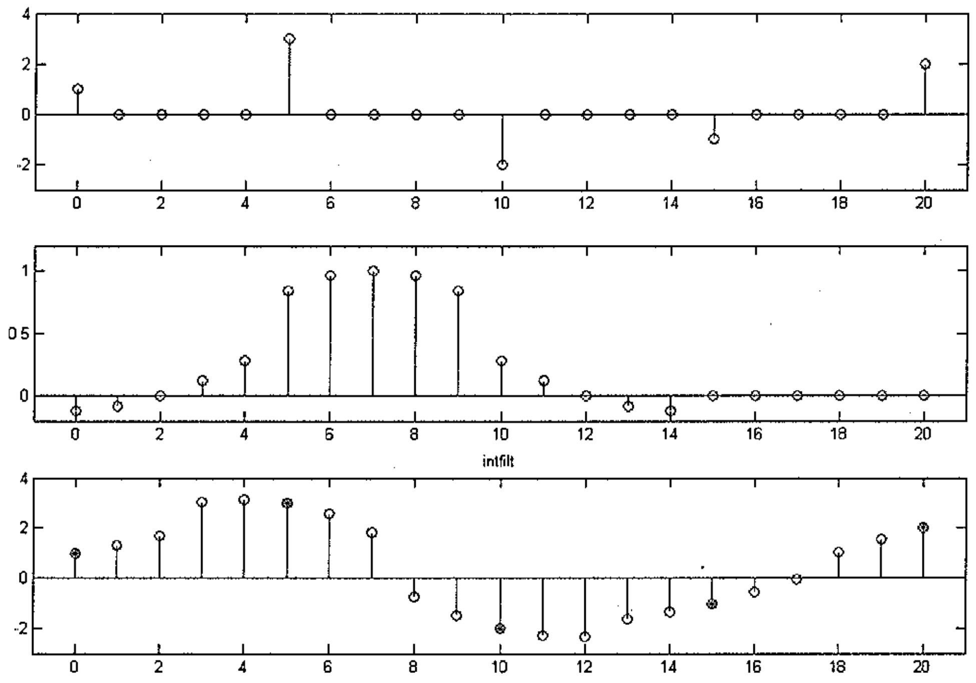

Fig. 2 Parabolic Interpolation: h=intfilt(5,2,'lagrange') in Matlab

The result in Fig. 2 is a bit strange-looking. In particular, we note that the impulse response (middle of Fig. 2) is not as smooth as we might like. Note the large jump between the 5th and the 6th samples. Further, this is a longer impulse response (length 15) as compared to the linear case (length 9). By considering the output (bottom of Fig. 2) we might suppose we have done better than the linear case, but not as well as we might have hoped. Indeed, we will need to look at the frequency responses of these filters to see what ***intfilt*** accomplishes.

To see why the impulse response for this parabolic case has large jumps, we can look at the way it is derived, and for this, we consider Fig. 3.

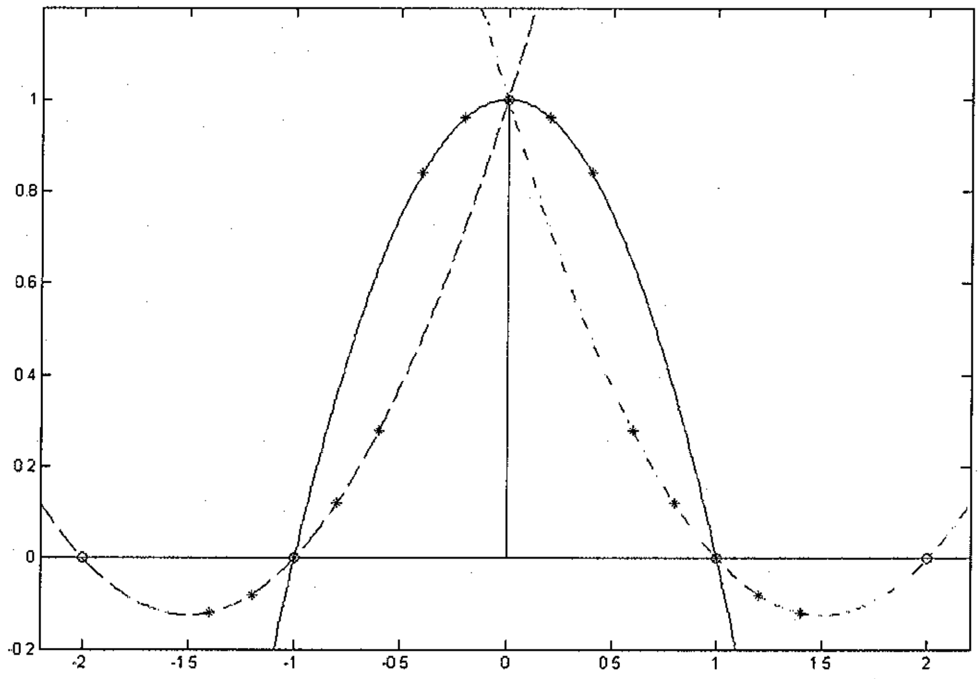

Fig. 3 Origin of the filter coefficients for parabolic interpolation

In Fig. 3, we have an impulse at zero (with corresponding zero values at all other integers – specifically at -2, -1, 1, and 2 shown here). As we fit parabolas to each three consecutive points, most of these parabolas (for example, the one that has samples at -10, -9, and -8) are all zero. There are only three non-zero parabolas – the ones shown in Fig. 3. The first of these (dashed) has the impulse on the right, the second (solid) has it in the middle, and the third (dot-dash) has it on the left. As with any polynomial curve fitting, we will trust the result only close to the middle sample. Indeed, the parabola (running to infinity) is clearly unsuited if we stray far from the center. Accordingly, for interpolation by 5, we choose impulse response value at -1.4, -1.2, -1, -0.8, and -0.6 from the first parabola (shown as stars). The middle of the second parabola is sampled at -0.4, -0.2, 0, 0.2, and 0.4. The third parabola is sampled at 0.6, 0.8, 1, 1.2, and 1.4. This procedure essentially describes the "trick" method of obtaining the impulse response. Notice the jump between the sample at -0.6 and the one at -0.4 where we also jump from one parabola to the other. Likely it occurs to the reader that we might better have just used the middle parabola from, say, -1 to +1. We will consider this in a moment, but this is NOT the way ***intfilt*** works.

TWO MORE CASES FOR COMPARISON

Before we move on to the minimum norm case, we want to look at two more impulse responses, both of which, like the linear case, are length 9. The first of these is the simple parabola, as suggested above. The interpolation result for this case is seen in Fig. 4. Note that the impulse response comes from the middle parabola of Fig. 3. The second new case is to choose the impulse response as the center lobe of a sinc function, and the interpolation result for this case is in Fig. 5.

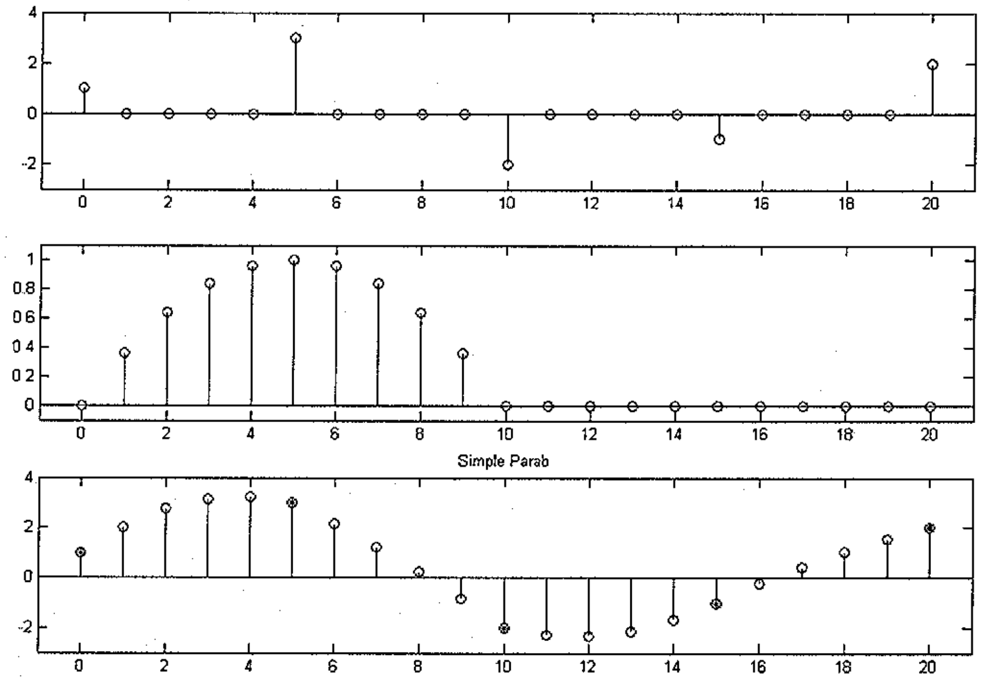

Fig. 4 Simple parabolic impulse response

When we look at these two impulse responses, we note that both are rounder or "fatter" than the linear case; the sinc being less fat than the parabola. Neither has the sharp jump we saw with the length 15 ***intfilt*** case. And, both of the interpolated results look smooth, although the parabolic case might appear just a little over rounded to readers who are accustomed to looking as sinusoidal waveforms. Indeed, we would tend to expect the sinc case to provide a baseline example.

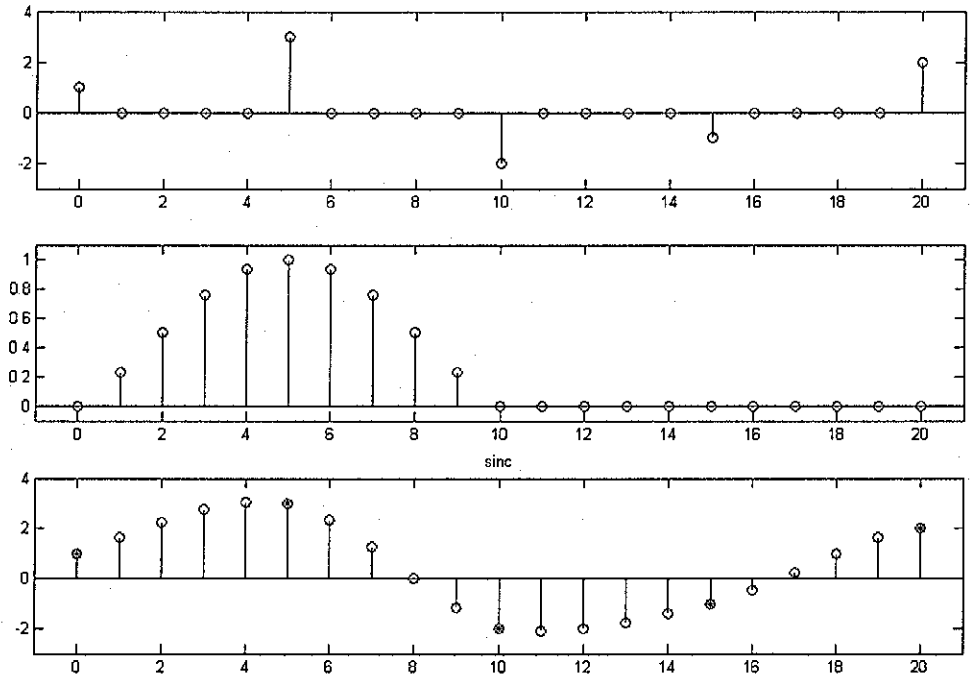

Fig. 5 Sinc lobe impulse response

THE MINIMUM NORM PARABOLA

The "minimum norm" parabola we want to use here is found by fitting a parabola to two points, subject to a minimum norm condition on the coefficients of the parabola. That is, as with linear interpolation, we only want to use two consecutive points. But this in itself is insufficient to define a parabola – until we impose the additional condition on the norm.

In contrast two Fig. 3 where we take three points at a time, and therefore, we have three cases of interest (three parabolas), here we have only two cases of interest (two parabolas): the case where the impulse is the first sample, and the case where it is the second sample (Fig. 6). Consider the case where we are fitting the points x(-1)=0 and x(0)=1 (solid curve of Fig. 6). We thus seek a parabola:

$$x(t) = at^2 + bt + c \tag{1}$$

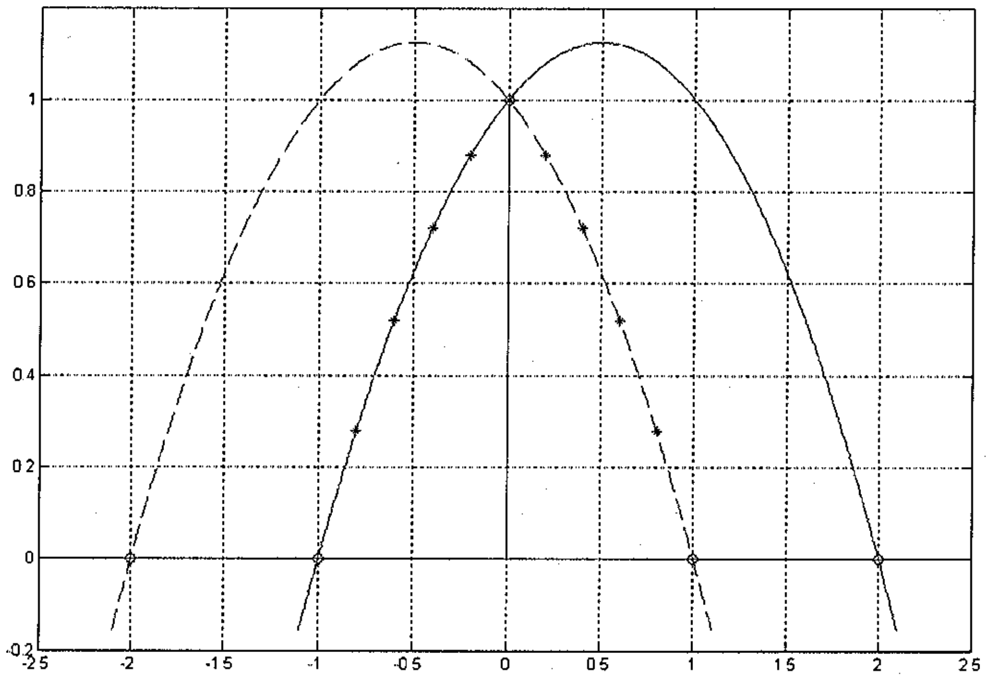

Fig. 6 An impulse and the two corresponding minimum norm parabolas

subject to the additional conditions:

$$x(-1) = 0 \tag{2a}$$

$$x(0) = 1 \tag{2b}$$

$$N^2 = a^2 + b^2 + c^2 \quad \text{(to be minimized)} \tag{2c}$$

These are easy to solve because equation (2b) immediately gives us c=1 and equation (2a) then gives us b = (a + 1). Substituting these into equation (2c) gives us:

$$N^2 = 2a^2 + 2a + 2 \tag{3}$$

and we can minimize N^2 as:

$$\partial N^2/\partial a = 4a + 2 = 0 \tag{4}$$

and finally:

$a = -(1/2)$ (5a)

$b = (1/2)$ (5b)

$c = 1$ (5c)

This gives the solid curve in Fig. 6, and by a similar procedure (or just by symmetry) we also get the second parabola (dashed in Fig. 6). Samples of these parabolas (stars) now gives us the length 9 impulse response:

$$h_{mn} = \{ 0.28 \quad 0.52 \quad 0.72 \quad 0.88 \quad 1.00 \quad 0.88 \quad 0.72 \quad 0.52 \quad 0.28 \} \qquad (6)$$

Use of this impulse response for our interpolation example is shown in Fig. 7.

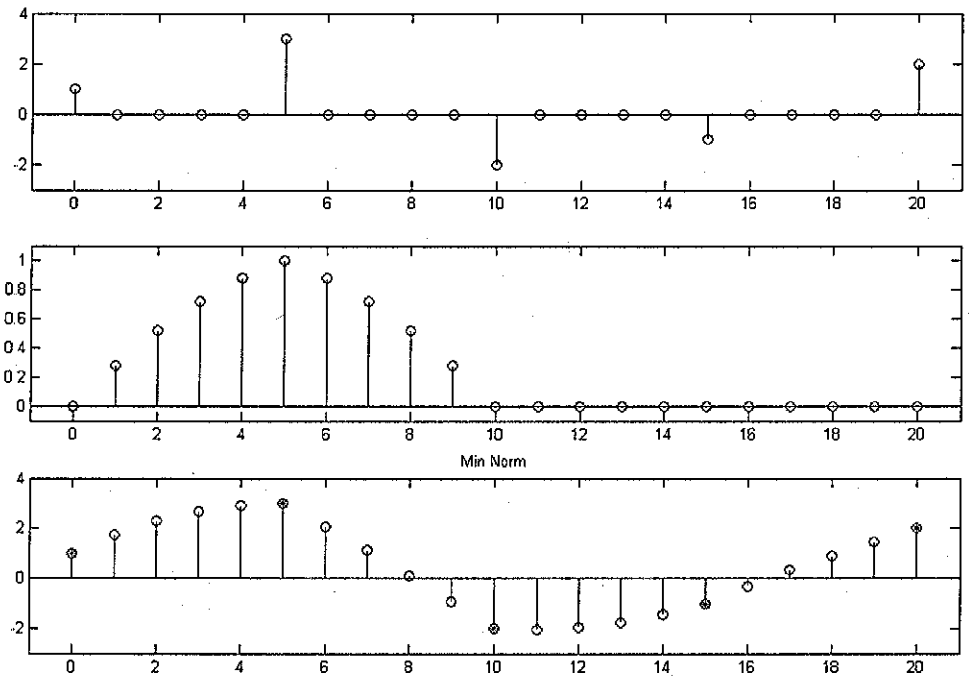

Fig. 7 Interpolation with minimum norm parabolas.

{ It is perhaps curious that the minimum norm parabola is not the straight line. We might have expected this. Note however that if it were a straight line, then a=0, the slope (b) would have to be of magnitude 1, and c=1, so the squared norm would be 2. The squared norm with the results of equations (5) is 1.5. }

Table 1 shows the impulse response values for the five cases we are comparing, and Fig. 8 presents the same results graphically:

TABLE 1

	Linear	Min Norm	Intfilt	Simple Parabola	Sinc
h(7)=h(-7)	0	0	-0.12	0	0
h(6)=h(-6)	0	0	-0.08	0	0
h(5)=h(-5)	0	0	0	0	0
h(4)=h(-4)	0.20	0.28	0.12	0.36	0.2339
h(3)=h(-3)	0.40	0.52	0.28	0.64	0.5046
h(2)=h(-2)	0.60	0.72	0.84	0.84	0.7568
h(1)=h(-1)	0.80	0.88	0.96	0.96	0.9355
h(0)	1.00	1.00	1.00	1.00	1.0000

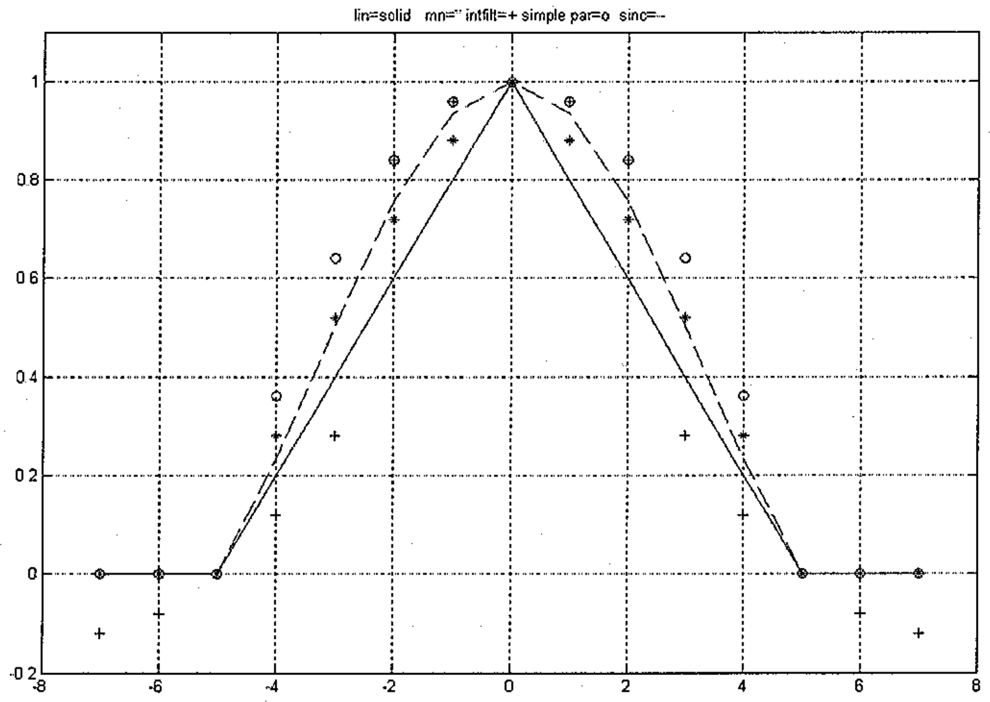

Fig. 8 The five impulse responses. Linear (solid line evaluated at the integers, sync (dashed line evaluated at integers), intfilt (+), simple parabola (o) and minimum norm (stars).

EVALUATIONS

We can make our first evaluations of the five different interpolation filters based on the five different output displays, Fig. 1, Fig. 2, Fig. 4, Fig. 5, and Fig. 7. From these, it is clear that the last three are superior to the first two. That is, any one of the three "fatter" length 9 responses seems better than either the linear, or the length 15 ***intfilt*** parabola. Beyond these generalizations, we need to make a more careful examination, and may find the results difficult to interpret. First we can look at the situation in the frequency domain, it terms of zero plots, and frequency responses.

ZERO PLOTS

Fig. 9 shows the pole/zero plots for the five interpolation filters. All the poles are at z=0, and are irrelevant to the current discussion. Note that in (a) and (b) we see that the unit circle zeros are all multiple order, a feature of the ***intfilt*** function. The other three all have eight zeros spread out, and are quite similar. The multiple-ordered zeros will result in very flat near zero stopband regions (see below).

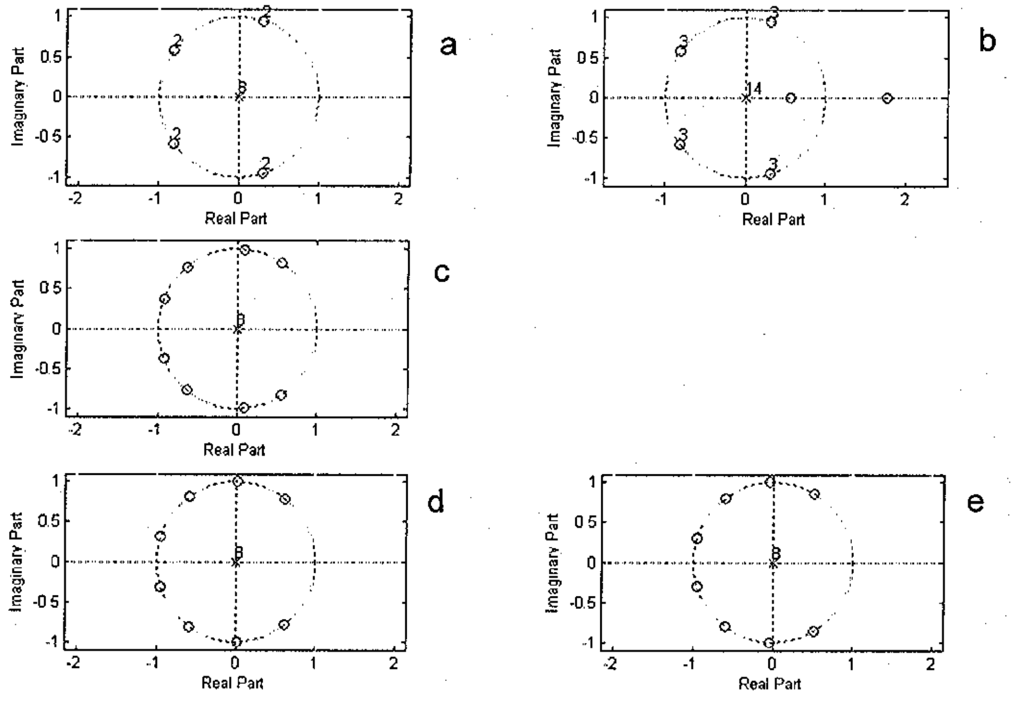

Fig. 9 Top: linear left, parabolic (intfilt) right. Middle (minimum norm). Bottom: simple parabola (left), sinc (right).

FREQUENCY RESPONSES

It is too messy to try to show all five frequency responses on the same graph, so we will instead show selected responses to compare. Fig. 10 shows the linear and parabolic (***intfilt***) cases.

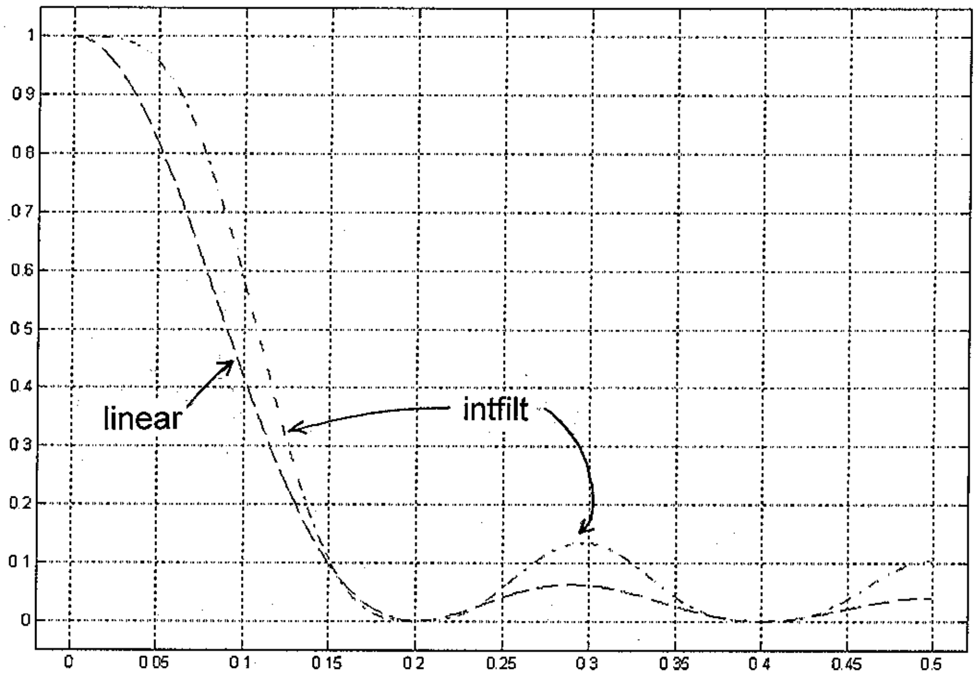

Fig. 10 Linear and ***intfilt*** parabolic cases.

As was clear from the zero plots, the two filters have unit circle zeros in the same places. We see that the corresponding frequency responses are flat about the zero frequencies, but tend to bump back up elsewhere. As for the passband, we do see that the ***intfilt*** response is a better low-pass in that it better approximates the desired cutoff region at a frequency of 1/10. It starts to hint at being rectangular.

Figures 11-13 compare the minimum norm to the linear, simple parabolic, and the sinc case. The most basic conclusion (based on frequency response appearance) is that minimum norm, parabolic, and sinc are all relatively similar, and are slightly better than linear, and substantially better than ***intfilt*** parabolic. So we are left with the same impression we had by looking at the time domain. The one thing we may need to be aware of is that the exact positions of the zeros may make a difference to additional tests, and that these differences may be artificial. Specifically, if we try to interpolate sinewaves, and if we use harmonic distortion as an error measure, one filter choice might strongly attenuate a particular harmonic and appear better simply as a result of the choice of test signal.

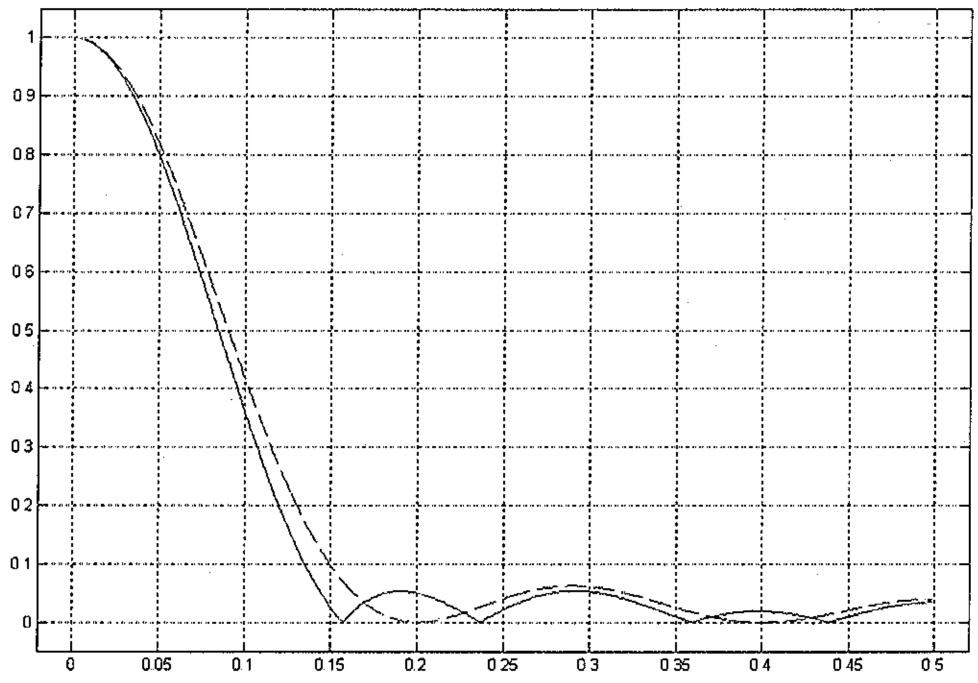

Fig. 11 Linear (- - -) vs. minimum norm (solid)

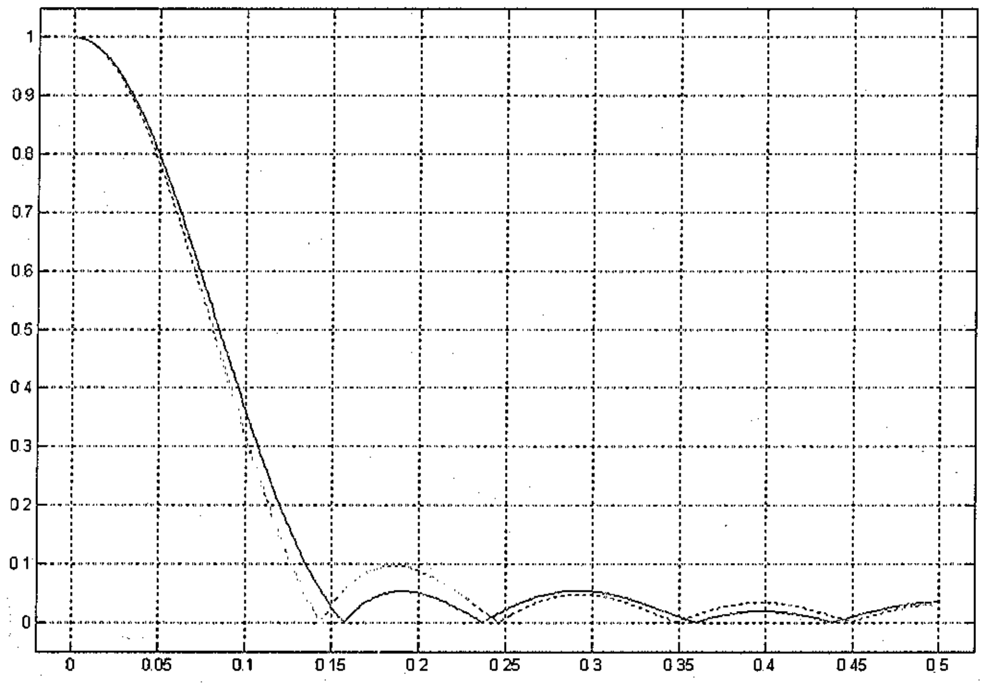

Fig. 12 Minimum norm (solid) vs. fat (simple) parabola (- - -)

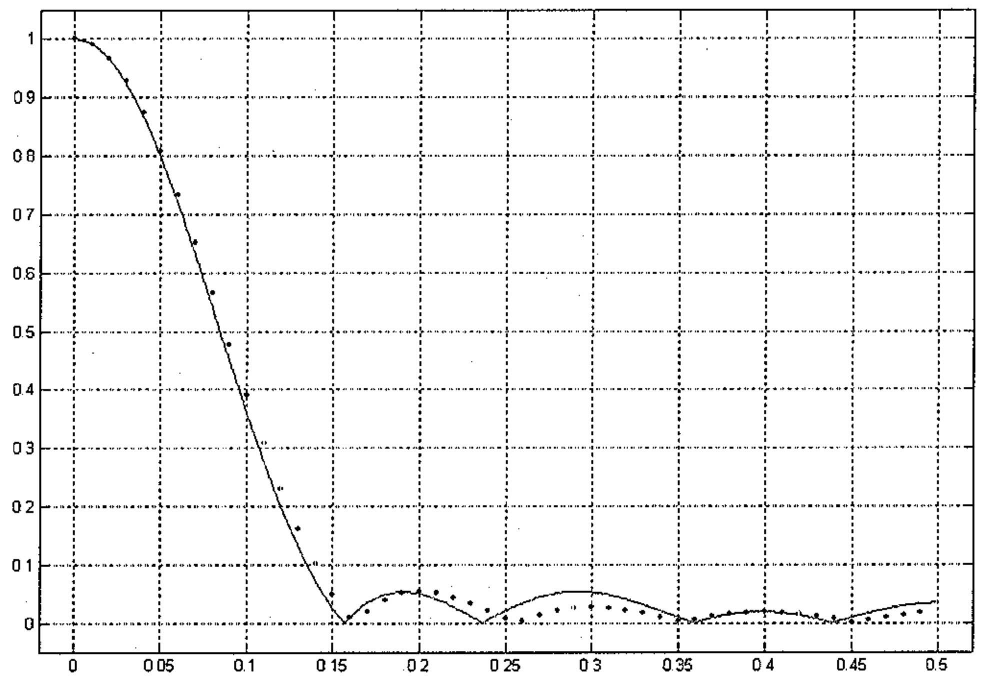

Fig. 13 Minimum norm (solid) vs. sinc (dots)

USE OF A TEST SIGNAL

The classical method of testing an interpolator is to begin with a suitable test signal, downsample it, interpolate it, and compare the interpolated values to the ones we removed. The success or failure of this depends on the properties of the test signal as well as the interpolation filter. Accordingly, we are really only indicating the general nature of the tests we try, and the results should not be judged to have a great deal of meaning unless the differences are fairly large (say 2:1). Further, the test signal here will be chosen so as to show how the choice of signal may bias the findings.

In particular, we can choose the test signal to be a sine wave of frequency 1/20 (20 samples per cycle). When we downsample this by 5, we still have four samples per cycle, which is more than enough (we needed more than 2 according to the sampling theorem). In fact (see Fig. 14), for this case the samples that are kept are

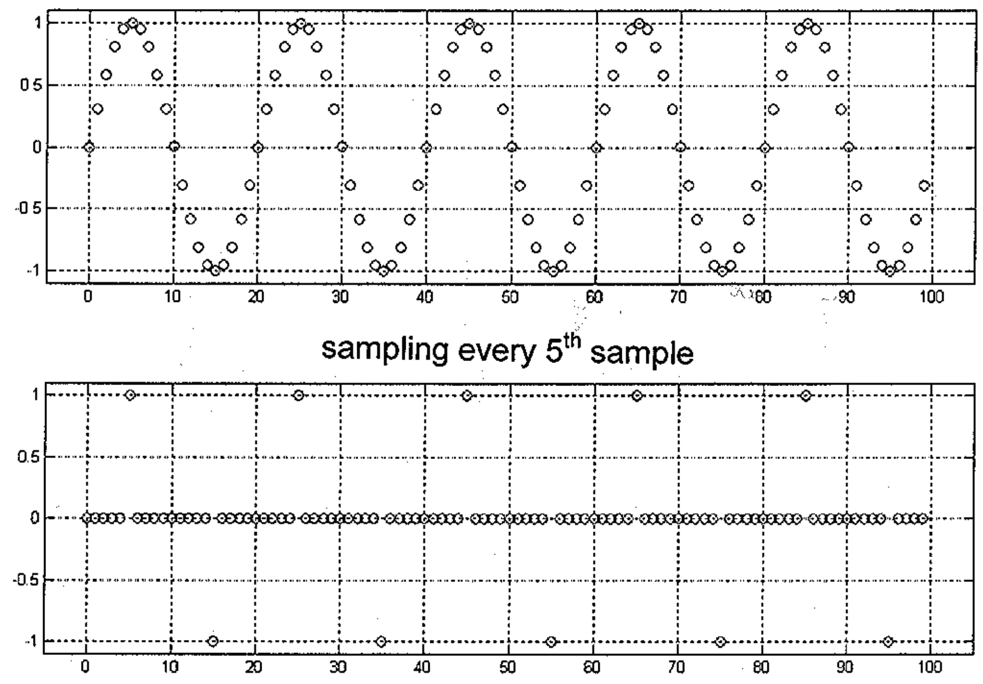

Fig. 14 Taking every fifth sample of the sinewave at the top produces only two non-zero samples (+1 and -1) per cycle.

the sequence .....0, 1, 0, -1..... and we can understand interpolation, in this case, as placing the impulse response of our choice for a length 9 interpolator about each of the +1 or -1 values. This means that the sinusoidal half-lobes are replaced by the impulse response of the interpolator. Accordingly, if the impulse were the sinusoidal half lobe (say samples 1-9 of the top portion of Fig. 14 – another choice for an interpolation filter in fact!), we would have perfect interpolation – for this particular input. This fairly unique choice allows us to discuss the interpolation errors we calculate in terms of how well the impulse response matches the sinusoidal lobe.

Fig. 15 shows an interpolation example using the simple parabolic case, and visually, the result looks good. It looks like what we expect for a sine wave. Note that the delay of four samples in the output is due to the linear phase delay (N-1)/2=4, which is the start-up transient. This transient would appear in an actual implementation or simulation, including the use of Matlab's ***filter*** function. Because of the exact periodicity of the input here and the finite filter length, we have no problem choosing any full cycle (20 consecutive samples) as a steady-state result. Fig. 16 and Fig. 17 show the corresponding results for the sinc and minimum norm cases. Both these appear to the eye to be slightly or somewhat "under-rounded."

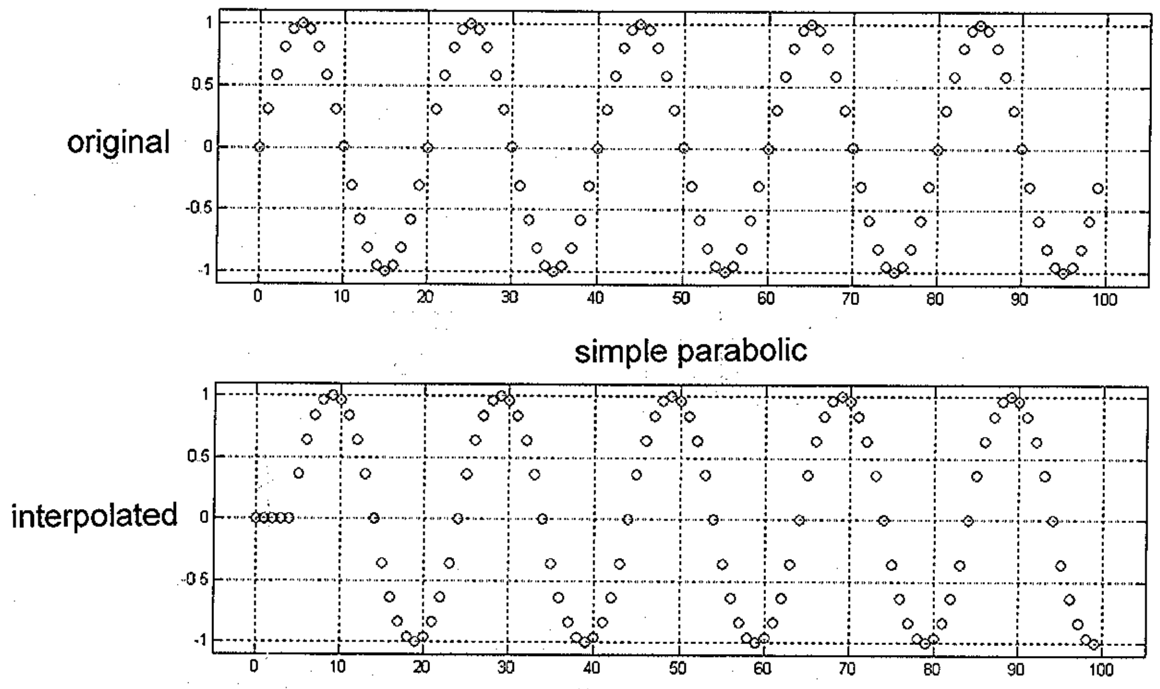

Fig. 15 Simple parabolic looks quite good

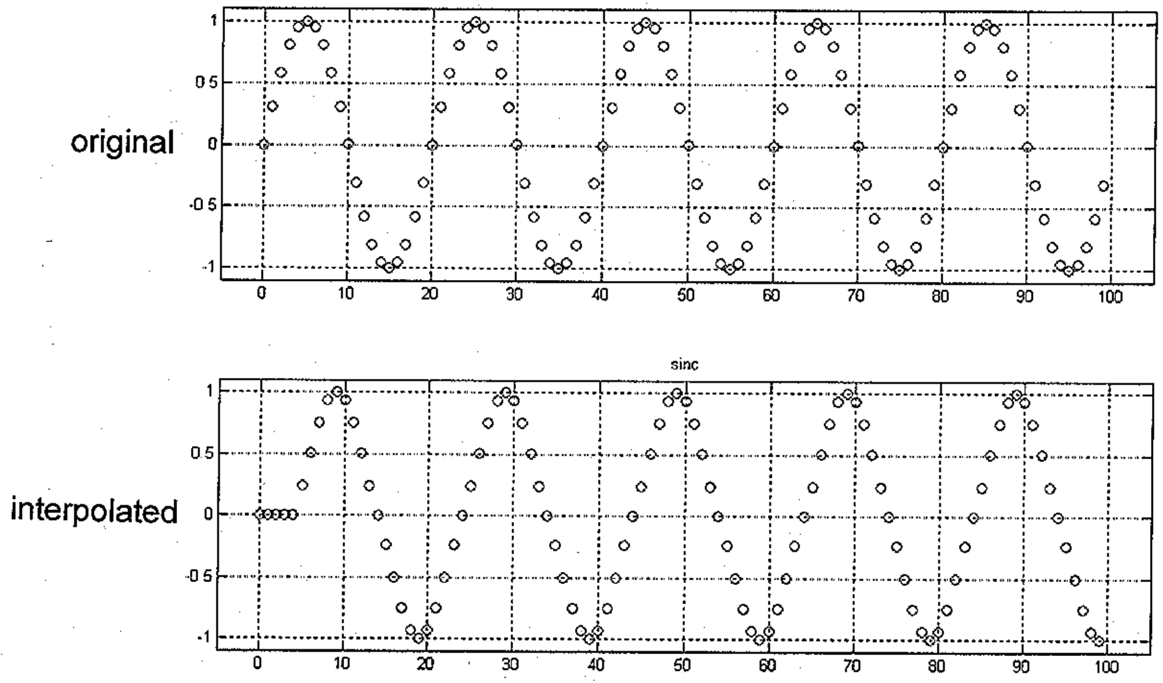

Fig. 16 Sinc interpolation – perhaps ever so under-rounded

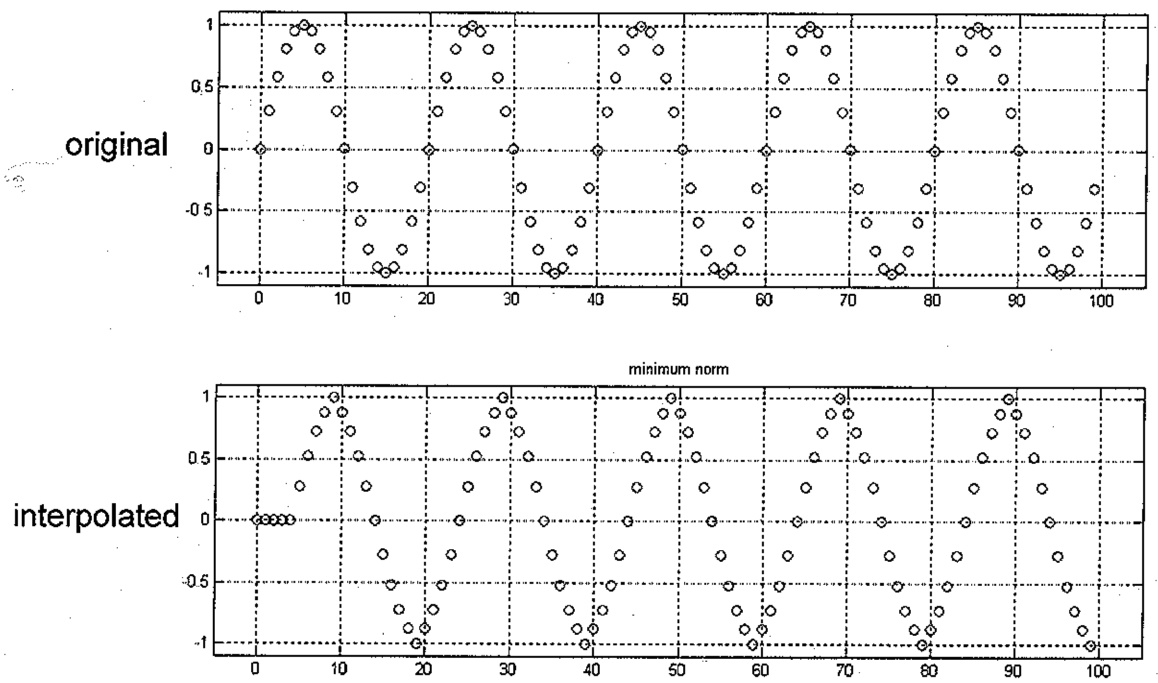

Fig. 17 Minimum norm interpolation looks under-rounded

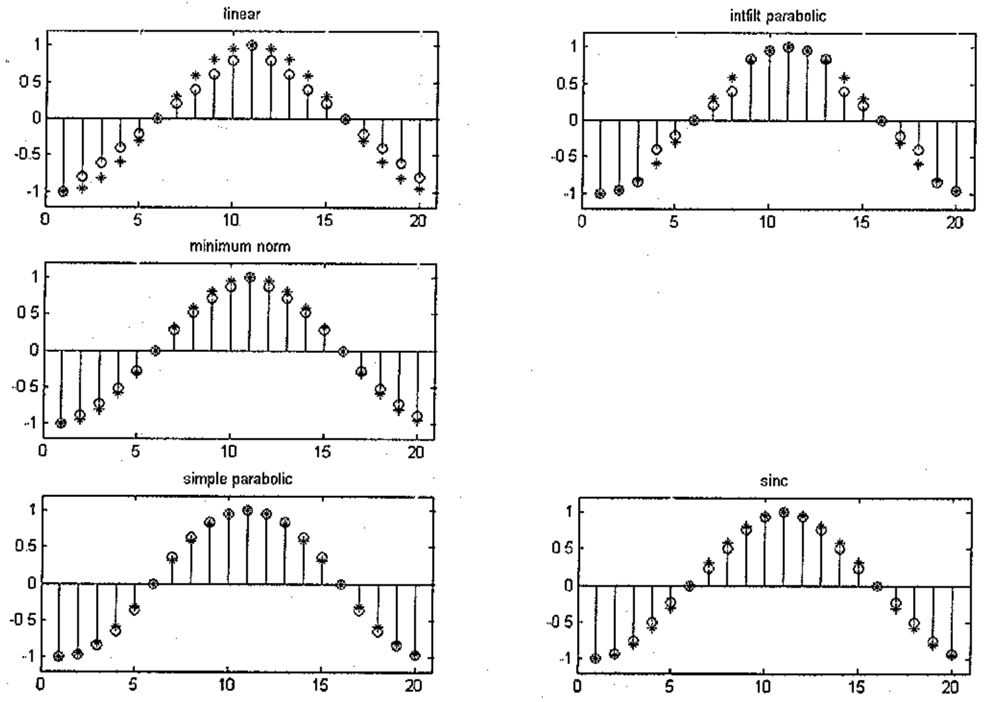

Fig. 18 Interpolations (circles) of single cycle (star = original)

Fig. 18 shows the shaping of all five impulse response we have been considering as open circles, relative to the actual sinewave lobe (stars). Here we are considering just a single cycle relative to figures 15-17. From these plots we note that linear and ***intfilt*** parabolic are not good choices. We also see that the simple parabolic is slightly over-rounded, while minimum norm and sinc are under-rounded. Minimum norm shows its under-rounding more near the top, while sinc shows its under-rounding more near the zero crossings. In fact, note the suggestion of a sharper triangular-like peak in Fig. 17. This we will discuss more below.

SQUARED ERROR AND HARMONIC DISTORTION

We can do a few more tests, based on our isolation of single cycles for our test case. Often squared error in the time domain is useful (or at least, expected), and in the case of the sinewave, we are interested in Total Harmonic Distortion (THD). It is clear how to compute the error on all 20 points, square it, and total it. Here we also will find it easy to compute the spectrum using the FFT since we have exactly 20 samples per cycle. We can look at the spectra (as in Fig. 19) and we can also compute the THD as:

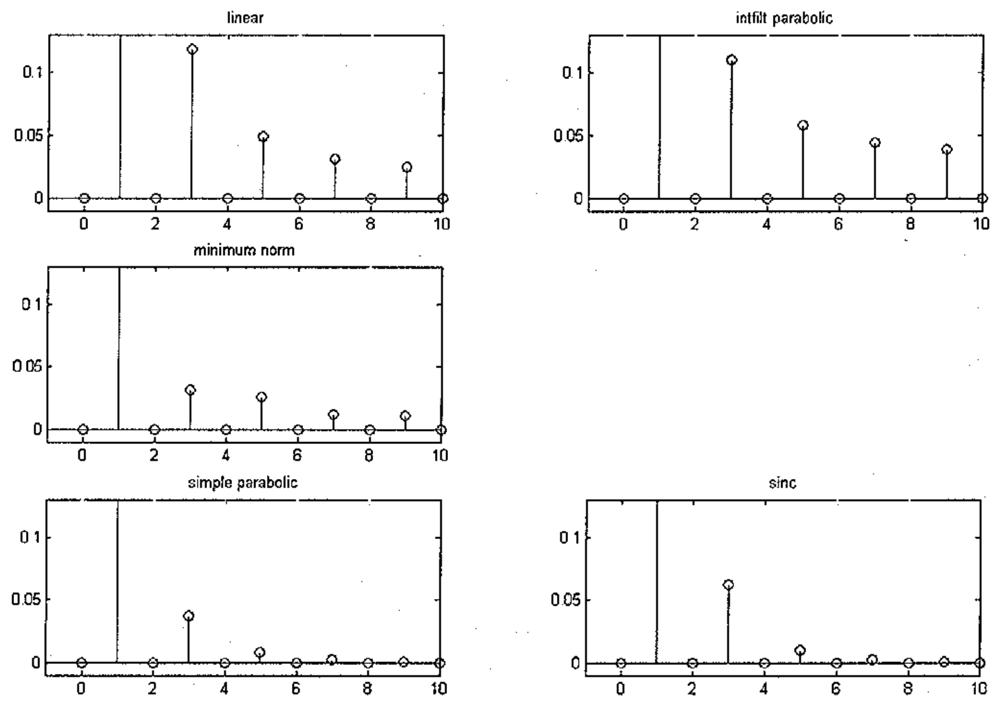

Fig. 17 Harmonic distortion tests. Here the spectral energy at frequency bin 1 (the fundamental) as normalized to 1, and is always off scale.

$$THD = [A_2^2 + A_3^2 + A_4^2 + \ldots]^{1/2} / A_1 \tag{7}$$

where the A's are the amplitudes of the harmonics. In this case, the signals are bandlimited by the FFT (to 10), and the symmetry means that all even harmonics are zero. We further normalize all the harmonic amplitudes by dividing by the amplitude of the fundamental. In order to see the amplitudes of the harmonics, we plot only from 0 to 0.13 so all the fundamentals are chopped off in Fig. 19. The results of the calculations are given in Table 2

TABLE 2

	THD	SQUARED ERROR
LINEAR	0.1344	0.4546
INTFILT PARABOLIC	0.1377	0.1928
SIMPLE PARABOLIC	0.0378	0.0255
SINC	0.0628	0.0622
MIN NORM	0.0444	0.0736
SINE LOBE	0	0

Table 2 strictly applies only to the exact sinewave interpolation problem, and we note that (aside from the result of the sine lobe which is perfect by definition), the simple parabolic shape is best in terms of both THD and squared error (notably outperforming sinc here). Minimum norm has the interesting property that its THD is only slightly higher (4.44%) than simple parabolic (3.78%) and is better than sinc (6.28%). This is despite the fact that the minimum norm shaping appears under-rounded (Fig. 18) as noted.

This might remind long-time readers of the differential amplifier triangle-to-sine converter which was often implemented by using the input non-linearity of an operational transconductance amplifier (OTA) [3]. Both theory and experiment indicated an optimal triangle input amplitude of 78mV. This produced the minimum THD. However, the resulting sinewave approximation looked, on an oscilloscope, to be under-rounded, and there was a noticeable residual "point" corresponding to the triangle peaks, much as we see in Fig. 17. Many manufacturers who used this approach chose to increase the triangle amplitude to as much as 100 mV, rounding out the peak so that the "sine" output looked more like what people expected, despite the clearly audible harmonic distortion.

REFERENCES:

[1] help file for Matlab's ***intfilt*** function and references there

[2] B. Hutchins, "Interpolator Based Filter Design," Electronotes, Vol. 20, No. 198, June 2001, pp 44-50

[3] B. Hutchins, "Mathematical Analysis of Differential Amplifier Triangle-to-Sine Converter," Electronotes, Vol. 9, No. 82, October 1977, pp 5-17